



# **Red Hat Enterprise Linux Kernel Device Drivers**

# Duration: 5 Days Course Code: RHD362

#### Overview:

Red Hat® Enterprise Linux® Kernel Device Drivers (RHD362) teaches experienced C programmers already familiar with the Linux kernel architecture the skills and strategies they need to develop device drivers. The course covers device driver models (including character, block, and network device drivers); device interaction (including port I/O, memory mapped I/O, interrupt handling, and DMA transfers); managing PCI and USB devices; strategies for deferring activity using tasklets and work queues; device registration using the Unified Device model and the sysfs filesystem; and process interaction, including basic file operations, polling, and wait queues.

### **Target Audience:**

Experienced C programmers with a good understanding of the Linux kernel who want to learn how to develop device drivers for Linux systems.

# **Objectives:**

Introduction and Review of Kernel Programming	<ul> <li>•</li> </ul>
•	Direct Memory Access
Device Drivers	• • • • • • • • • • • • • • • • • • •
•	PCI Drivers
Unified Device Model	• • • • • • • • • • • • • • • • • • •
•	USB Drivers
Interrupt Handling	• • • • • • • • • • • • • • • • • • •
•	Introduction to Network Device Drivers
Advanced File Operations	• • • • • • • • • • • • • • • • • • •
•	Introduction to Block Device Drivers
Interacting With Devices	

## Prerequisites:

- Experience in C programming
- Red Hat Enterprise Linux Kernel Internals (RHD361) or equivalent experience

# Content:

# Introduction and Review of Kernel Programming

- \* What is a Kernel?
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports

**RHD362** 

- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function

- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers

**USB** Drivers

The Universal Serial Bus

Linux Kernel Variants

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules
 More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystemDevice Model Structures

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

I/O Ports

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

00 966 92000 9278

training@globalknowledge.com.sa

Initializing DMA

Deferred Activity: Work Queues

Registering File Operations

Netlink Sockets and User Events

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel
 Kernel Implementation

- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
   System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

- Lockless IOCTLs
- Sleeping
- Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

www.globalknowledge.com/en-sa/

Releasing a PCI Device

Managing PCI Resources

Accessing Configuration Space

Initializing DMA

Wait Queues

Polling

I/O Ports

- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures

- USB Descriptors
- USB Transfer Types
   Linux USB Implementation

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Netlink Sockets and User Events

00 966 92000 9278

File I/O Conventions
 The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

training@globalknowledge.com.sa

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Control

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

- USB Drivers
- USB Registration
   Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device RegistrationMiscellaneous Devices

Handling Interrupts

Interrupt Structures

Deferred Activity: Tasklets

ISR Registration

System Context

Softirqs

www.globalknowledge.com/en-sa/

Registering File Operations

Interacting with Character Device Drivers

Interacting with Block Device Drivers
 Interacting with Network Interface Cards

Character Device Drivers: Registration

Device Driver Design

Device Nodes

File Operations

- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues

- Initialization Example
- Handling Requests

Deferred Activity: Work Queues

ISR Registration

System Context

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Request Blocks

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces
 USB Class Drivers

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Registration

The gendisk structure
 Request Queues

Initialization Example

Handling Requests

training@globalknowledge.com.sa

Block Device Driver Implementation

00 966 92000 9278

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Softirgs

- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
   Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure
 Request Queues

Initialization Example

Linux Kernel Variants
 Interacting with the Kernel

www.globalknowledge.com/en-sa/

Handling Requests

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

USB Class Drivers
 USB Request Blocks

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors

- USB Transfer Types
- Linux USB Implementation

- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree
 Writing Kernel Modules

Interacting with the Kernel

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

training@globalknowledge.com.sa

Managing PCI Resources

Accessing Configuration Space

00 966 92000 9278

Initializing DMA

Polling

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tving it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Device Driver Design

Device Nodes

File Operations

- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs

Sleeping

Polling

I/O Ports

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Drivers

www.globalknowledge.com/en-sa/

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Accessing Configuration Space

Initializing DMA

- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
   Lockless IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs

- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions

- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks
 Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Request Queues
 Initialization Example

Handling Requests

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Device Model Structures
 Tying it all Together

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Deferred Activity: Work Queues

00 966 92000 9278

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirqs

training@globalknowledge.com.sa

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

USB Drivers

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

- Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

www.globalknowledge.com/en-sa/

Implementing IOCTLs

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Registering File Operations

File Operations

- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers

Character Device Drivers: Registration

Interacting with Network Interface Cards
 Device Nodes

- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree

- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules

- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()

Example: Serial Mouse

Implementing IOCTLs

Variations on wait event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices
 USB Descriptors

USB Transfer Types

USB Drivers

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Request Queues
 Initialization Example

Handling Requests

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

training@globalknowledge.com.sa

Interacting with the Kernel

00 966 92000 9278

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

Pollina

I/O Ports

- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
   PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information

Control

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

- Network Device Callback Functions
- Network Driver Initialization
   Network Driver Transmission Queue

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Advanced File Operations

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Interacting with the Kernel

Handling Requests

IOCTLs

www.globalknowledge.com/en-sa/

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirgs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information

- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

USB Request Blocks

USB Drivers

training@globalknowledge.com.sa

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

00 966 92000 9278

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Lockless IOCTLsSleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

USB Request Blocks

USB Drivers

www.globalknowledge.com/en-sa/

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O

- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions

Using USB request blocks

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

**Block Device Registration** 

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Model Structures

Userspace event agents

**Device Registration** 

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL ConventionsImplementing IOCTLs

Lockless IOCTLs

Sleeping

training@globalknowledge.com.sa

Wait Queues

Deferred Activity: Work Queues

00 966 92000 9278

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

**Block Device File Operations** 

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node ProblemFile Operations

Registering File Operations

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

www.globalknowledge.com/en-sa/

Wait Queues

Implementing IOCTLs

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes

- Character Device Drivers: Registration
- The Device Node Problem

- Wait Queues
- Variations on wait\_event()

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

**Block Device Registration** 

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

training@globalknowledge.com.sa

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

Interacting with Character Device Drivers

00 966 92000 9278

Interacting with the Kernel

Handling Requests

Request Queues

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

Pollina

I/O Ports

Variations on wait event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

www.globalknowledge.com/en-sa/

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

Interacting with Character Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Block Device Driver Implementation

- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control

- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission

- Interacting with Block Device Drivers
- Interacting with Network Interface Cards

Interacting with Block Device Drivers

Device Nodes

File Operations

Interacting with Network Interface Cards

Character Device Drivers: Registration

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices
 Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

training@globalknowledge.com.sa

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

00 966 92000 9278

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Registering File Operations

- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR RegistrationSystem Context
- Softirgs
- = Solurgs
- Deferred Activity: Tasklets
   Deferred Activity: Work Queues
- Deferred Activity: work Que
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping

I/O Ports

- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Multiplexed I/O
Polling

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

www.globalknowledge.com/en-sa/

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Accessing Configuration Space

Initializing DMA

- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA

Coherent Allocations

Network Driver Transmission Queue Control Network Driver Transmission Queue

Control

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues
 Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

training@globalknowledge.com.sa

Wait Queues

Wait Queues

Multiplexed I/O

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

00 966 92000 9278

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O ConventionsThe sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

**Block Device File Operations** 

Block Device Driver Implementation

- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: RegistrationThe Device Node Problem
- File OperationsDefining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

www.globalknowledge.com/en-sa/

Wait Queues

Wait Queues

Multiplexed I/O

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Deferred Activity: Work Queues

Netlink Sockets and User Events

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests

### **Device Drivers**

- Device Drivers
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations

**RHD362** 

- Defining File Operations
- Registering File Operations
- File I/O Conventions

- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers

I/O Ports

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices
 Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

training@globalknowledge.com.sa

Compiling Kernel Modules

The Device Node Problem

More about Kernel Modules
 Review of Kernel Structures

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

00 966 92000 9278

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

Accessing I/O Memory Buffers

- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

www.globalknowledge.com/en-sa/

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Block Device File Operations

Block Device Driver Implementation

- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception

- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions

- Defining File Operations
- Registering File Operations

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration
 Initializing USB Interfaces

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

training@globalknowledge.com.sa

Control

USB Request Blocks

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization
 Network Driver Transmission Queue

Socket Buffers (Packets)

Network Device Callback Functions

00 966 92000 9278

USB Drivers

Managing PCI Resources

Linux USB Implementation

Accessing Configuration Space

Initializing DMA

Pollina

I/O Ports

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Registering File Operations

- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
   Initializing DMA

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

www.globalknowledge.com/en-sa/

Control

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports

- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers

- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation

Reception

I/O Scheduling

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

00 966 92000 9278

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

I/O Ports

training@globalknowledge.com.sa

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Initializing DMA

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
   Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

I/O Ports

www.globalknowledge.com/en-sa/

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Initializing DMA

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

- Device Registration
- Miscellaneous Devices
- Handling Interrupts
   Interrupt Structures

ISR Registration

System Context

Softirgs

- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts

Streaming DMA Mappings

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File OperationsDefining File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Registering File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

training@globalknowledge.com.sa

Device Model Structures

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

00 966 92000 9278

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Synchronous USB Requests

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

The sysfs filesystem

Tying it all Together

www.globalknowledge.com/en-sa/

Registering File Operations
 File I/O Conventions

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues

- Initialization Example
- Handling Requests

Netlink Sockets and User Events

Netlink Sockets and User Events

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

I/O Ports

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration
 Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

training@globalknowledge.com.sa

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator FunctionsBlock Device Driver Implementation

00 966 92000 9278

Block Device Registration

Control

USB Request Blocks
 Using USB request blocks

USB Drivers

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Synchronous USB Requests

Network Device Callback Functions Network Driver Initialization

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softiras

- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirgs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe FunctionInitializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration
 Initializing USB Interfaces

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

www.globalknowledge.com/en-sa/

Control

USB Request Blocks

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device Registration

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

USB Drivers

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Accessing Configuration Space

- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors

USB Transfer Types

Block Device File Operations

Block Device File OperationsThe gendisk structure

Introduction to Network Device Drivers

Request Queues

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

I/O Ports

training@globalknowledge.com.sa

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

00 966 92000 9278

PCI Configuration Space

Initializing DMA

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Networking

- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
   Miscellaneous Devices
- Handling Interrupts
   Interrupt Structures
- ISR Registration
- System Context
- Softirgs
- Deferred Activity: Tasklets
   Deferred Activity: Work Queues

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

I/O Ports

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

www.globalknowledge.com/en-sa/

Initializing DMA

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs

- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse

- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Class Drivers

USB Request Blocks

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Netlink Sockets and User Events

00 966 92000 9278

File I/O Conventions

The sysfs filesystem

Tying it all Together

training@globalknowledge.com.sa

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Control

USB Descriptors

USB DriversUSB Registration

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
   Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

www.globalknowledge.com/en-sa/

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

The Linux Source Tree
 Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules

Compiling Kernel Modules

- ISR Registration
- System Context
- Softirgs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Sleeping

Polling

I/O Ports

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

training@globalknowledge.com.sa

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

00 966 92000 9278

Control

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Implementing IOCTLsLockless IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
   Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Network Driver Transmission Queue

USB Drivers
 USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Handling Requests

www.globalknowledge.com/en-sa/

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Control

USB Request Blocks

- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers

- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests

- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers

Request Queues

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

training@globalknowledge.com.sa

Releasing a PCI Device

Managing PCI Resources

00 966 92000 9278

Initializing DMA

Polling

I/O Ports

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
   Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

www.globalknowledge.com/en-sa/

Managing PCI Resources

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues

- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls

Linux USB Implementation

Accessing Configuration Space

Listing PCI Devices

USB Transfer Types

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices
 Handling Interrupts

Netlink Sockets and User Events

00 966 92000 9278

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Interrupt Structures

ISR Registration

System Context

training@globalknowledge.com.sa

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Schedulina

Reception

Control

USB Request Blocks

USB Drivers

- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests

#### Interacting With Devices

- Device Interaction
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

- Compiling Kernel Modules
- More about Kernel Modules
   Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

www.globalknowledge.com/en-sa/

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes

Character Device Drivers: Registration

- Softirqs
- Deferred Activity: Tasklets

Softiras

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Descriptors
 USB Transfer Types

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure
 Request Queues

Initialization Example

Handling Requests

training@globalknowledge.com.sa

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

00 966 92000 9278

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
   PCI Configuration Space
- PCI Configuration Space
  PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Accessing Configuration Space
   Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Registration

The gendisk structure

Initialization Example

Handling Requests

www.globalknowledge.com/en-sa/

Request Queues

Block Device Driver Implementation

Control

USB Request Blocks

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Network Driver Transmission Queue

- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API

- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O
 Polling

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Drivers

training@globalknowledge.com.sa

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

00 966 92000 9278

Accessing Configuration Space

Initializing DMA

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

- Interacting with Network Interface CardsDevice Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial MouseIOCTL Conventions

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Lockless IOCTLs

Sleeping

Polling

I/O Ports

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Drivers

www.globalknowledge.com/en-sa/

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Accessing Configuration Space

Initializing DMA

- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports

- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA

- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The aendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Deferred Activity: Work Queues

00 966 92000 9278

Netlink Sockets and User Events

File I/O ConventionsThe sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts
 Interrupt Structures

ISR Registration

System Context

Softirgs

training@globalknowledge.com.sa

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

www.globalknowledge.com/en-sa/

Registering File Operations

Device Driver Design

Device Nodes

File Operations

- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
   Interacting with Network Interface Cards

Character Device Drivers: Registration

- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests

#### Linux Kernel Variants

- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together

**RHD362** 

Implementing IOCTLs

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

Releasing a PCI Device

Managing PCI Resources
 Accessing Configuration Space

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants
 Interacting with the Kernel

Kernel ImplementationThe Linux Source Tree

Writing Kernel Modules

training@globalknowledge.com.sa

Compiling Kernel Modules

00 966 92000 9278

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

Initializing DMA

Pollina

I/O Ports

- Lockless IOCTLs
   Sleeping
- Sleeping
   Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Pollina
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
   Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Devices
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

www.globalknowledge.com/en-sa/

Compiling Kernel Modules

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

- More information
- Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

Network Device Callback Functions

Network Driver Transmission Queue

- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception

- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations

- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

Svstem Context

Softiras

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Class Drivers

USB Request Blocks

USB Descriptors

USB Drivers
 USB Registration

training@globalknowledge.com.sa

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

00 966 92000 9278

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Device Driver Design

Device Nodes

File Operations

- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

- Lockless IOCTLs
- Sleeping

Polling

I/O Ports

Wait QueuesWait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

USB Request Blocks

USB Drivers

www.globalknowledge.com/en-sa/

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Accessing Configuration Space

Initializing DMA

- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device

- More information
- Network Devices
- Network Device Callback Functions

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Request Queues

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

**Device Registration** 

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

Variations on wait\_event()

00 966 92000 9278

IOCTL ConventionsImplementing IOCTLs

Lockless IOCTLs

Sleeping

training@globalknowledge.com.sa

Wait Queues

Wait Queues

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Reception

Control

Network Device Callback Functions

Network Driver Transmission Queue

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device Registration

Block Device File Operations
 The gendisk structure

Block Device Driver Implementation

- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
   Registering File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

www.globalknowledge.com/en-sa/

Wait Queues

Wait Queues

Implementing IOCTLs

Variations on wait\_event()

Deferred Activity: Work Queues

Netlink Sockets and User Events

- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests

#### Unified Device Model

- Device Model Objects
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration

**RHD362** 

Miscellaneous Devices

- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Multiplexed I/O
 Polling

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

training@globalknowledge.com.sa

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

Interacting with Character Device Drivers

Interacting with Network Interface Cards

00 966 92000 9278

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

I/O Ports

- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Handling Requests

**Direct Memory Access** 

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

www.globalknowledge.com/en-sa/

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

Interacting with the Kernel

Managing DMA

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

- Socket Buffer API
- Transmission
   Reception

I/O Scheduling

- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests

- Device Driver Design
- Interacting with Character Device Drivers

Device Nodes

File Operations

Character Device Drivers: Registration

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures
 ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Drivers

USB Registration

USB Class Drivers

More information

Network Devices

Control

training@globalknowledge.com.sa

USB Request Blocks

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Network Driver Transmission Queue

00 966 92000 9278

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations

- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping

Polling

I/O Ports

- Wait Queues
- Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

www.globalknowledge.com/en-sa/

USB Request Blocks

USB Drivers

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Accessing Configuration Space

Initializing DMA

Variations on wait\_event()Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors

Network Device Callback Functions

Network Statistics
 Socket Buffers (Packets)

Socket Buffer API

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

00 966 92000 9278

IOCTL Conventions

Lockless IOCTLs

Sleeping
Wait Queues

Wait Queues

Multiplexed I/O
 Polling

I/O Memory Buffers

I/O Ports

training@globalknowledge.com.sa

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Registering File Operations

Interacting with Character Device Drivers
 Interacting with Block Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Transmission

I/O Scheduling

Reception

- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

www.globalknowledge.com/en-sa/

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O ConventionsThe sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs

- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues

- Polling
- Managing Exclusive Access

Accessing I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device
 Managing PCI Resources

Listing PCI Devices

USB Transfer Types

Linux USB Implementation

USB Descriptors

USB RegistrationInitializing USB Interfaces

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

Using USB request blocks

Svnchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Request Queues
 Initialization Example

Handling Requests

Block Devices

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

training@globalknowledge.com.sa

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

00 966 92000 9278

Interacting with Block Device Drivers

Interacting with the Kernel

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Introduction to Block Device Drivers

Network Device Callback Functions

Network Driver Transmission Queue

USB Drivers

Accessing Configuration Space

Initializing DMA

- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device File OperationsThe gendisk structure

Block Device Registration

Block Device Driver Implementation

- Socket Buffer API
- Transmission

I/O Scheduling

Request Queues

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

www.globalknowledge.com/en-sa/

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Reception

- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules

The Device Node Problem

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices
 Releasing a PCI Device

Listing PCI Devices

USB Descriptors
 USB Transfer Types

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Control

training@globalknowledge.com.sa

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Network Device Callback Functions

Network Driver Transmission Queue

00 966 92000 9278

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts
 Interrupt Structures

ISR Registration

System Context

Softirgs

Registering File Operations

File Operations

- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR RegistrationSystem Context
- Softirgs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device
 Managing PCI Resources

Listing PCI Devices

USB Transfer Types

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Network Device Callback Functions

Network Driver Transmission Queue

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Control

www.globalknowledge.com/en-sa/

USB Request Blocks

USB Drivers

Accessing Configuration Space

- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers

- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks

- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation

Socket Buffer API

Example Implementations

Block Device Registration

The aendisk structure

Request Queues
 Initialization Example

Handling Requests

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Polling

I/O Ports

training@globalknowledge.com.sa

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

00 966 92000 9278

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Transmission

I/O Scheduling

Reception

- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests

#### PCI Drivers

- Peripheral Component Interconnect
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

www.globalknowledge.com/en-sa/

Wait Queues

Wait Queues

Multiplexed I/O

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Deferred Activity: Work Queues

Netlink Sockets and User Events

- Registering File Operations
- File I/O Conventions
   The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping

- Wait Queues
- Wait Queues
- Variations on wait\_event()

- I/O Ports
- I/O Memory Buffers

Initializing DMA

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types
 Linux USB Implementation

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Interacting with the Kernel
 Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File OperationsDefining File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Registering File Operations

File I/O Conventions

The sysfs filesystem

training@globalknowledge.com.sa

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

00 966 92000 9278

Interacting with Block Device Drivers

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

USB Drivers

Managing PCI Resources

Accessing Configuration Space

- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
   Listing PCI Devices
- USB Descriptors
- USB Descriptors
   USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission

I/O Scheduling

- Reception
- Example Implementations

Block Device Registration

The gendisk structure
 Request Queues

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

www.globalknowledge.com/en-sa/

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes

- Defining File Operations
- Registering File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

Implementing IOCTLs

Variations on wait event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

IOCTL Conventions

Lockless IOCTLs

Sleeping

Pollina

I/O Ports

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types
 Linux USB Implementation

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Control
Network Statistics

USB Request Blocks

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

00 966 92000 9278

Socket Buffer API

TransmissionReception

I/O Scheduling

training@globalknowledge.com.sa

Network Device Callback Functions

Network Driver Transmission Queue

USB Drivers

Releasing a PCI Device

Managing PCI Resources

Accessing Configuration Space

Initializing DMA

Deferred Activity: Work Queues

Netlink Sockets and User Events

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softiras

- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
   PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics
 Socket Buffers (Packets)

Socket Buffer API

Transmission

www.globalknowledge.com/en-sa/

Control

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics

Socket Buffers (Packets)

- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation

Block Device Driver Implementation

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants
 Interacting with the Kernel

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

I/O Ports

training@globalknowledge.com.sa

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

00 966 92000 9278

Initializing DMA

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Handling Requests

Request Queues

Block Device File Operations

- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
   Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
   Device Registration

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

I/O Ports

www.globalknowledge.com/en-sa/

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Initializing DMA

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports

- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers

Streaming DMA Mappings

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information
 Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations
 Registering File Operations

File I/O Conventions

The sysfs filesystemDevice Model Structures

Tying it all Together

Device Registration

training@globalknowledge.com.sa

Userspace event agents

Miscellaneous Devices

Netlink Sockets and User Events

00 966 92000 9278

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

File I/O Conventions

The sysfs filesystem

Tying it all Together

www.globalknowledge.com/en-sa/

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests

#### Interrupt Handling

- Interrupt Context
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations

**RHD362** 

Defining File Operations

Netlink Sockets and User Events

Handling Interrupts

Interrupt Structures

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI DriversDriver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Request Queues

training@globalknowledge.com.sa

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

00 966 92000 9278

Control

USB Request Blocks

USB Drivers

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocksSynchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

ISR Registration

System Context

Softirgs

- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
   Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
   Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
   USB Descriptors

USB Transfer Types

USB Drivers

USB Registration

USB Request Blocks

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

www.globalknowledge.com/en-sa/

Control

Linux USB Implementation

Initializing USB Interfaces
 USB Class Drivers

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device Registration

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception

Example Implementations

Block Device File Operations

Handling Requests

Linux Kernel Variants

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI DriversDriver Probe Function

Initializing PCI devices
 Releasing a PCI Device

Listing PCI Devices

training@globalknowledge.com.sa

Managing PCI Resources

Accessing Configuration Space

00 966 92000 9278

Initializing DMA

Polling

I/O Ports

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel
 Kernel Implementation

- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Polling

I/O Ports

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

www.globalknowledge.com/en-sa/

Initializing DMA

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Deferred Activity: Work Queues

- Handling Interrupts
- Interrupt Structures
   ISR Registration
- System Context

Softirgs

- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports

- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space

- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network StatisticsSocket Buffers (Packets)

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

USB Transfer Types

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization
 Network Driver Transmission Queue

Example Implementations

Block Device Registration

The gendisk structure

Request Queues
 Initialization Example

Handling Requests

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

**Device Registration** 

Interrupt Structures

ISR Registration

System Context

Softirqs

training@globalknowledge.com.sa

Miscellaneous DevicesHandling Interrupts

Deferred Activity: Tasklets

00 966 92000 9278

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Registering File Operations

Interacting with Character Device Drivers

Interacting with Block Device Drivers
 Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with the Kernel

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
   Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

www.globalknowledge.com/en-sa/

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

The Linux Source TreeWriting Kernel Modules

Device Driver Design

Device Nodes

File Operations

- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration

- ISR Registration
- System Context
- Softirgs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues

Deferred Activity: Work Queues

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O
Polling

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

training@globalknowledge.com.sa

Interacting with the Kernel

00 966 92000 9278

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

I/O Ports

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
   Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Handling Requests

www.globalknowledge.com/en-sa/

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

USB Drivers
 USB Registration

- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues

Initialization Example

- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirgs

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Example: Serial Mouse
 IOCTL Conventions

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Lockless IOCTLs

Sleeping

Polling

I/O Ports

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types
 Linux USB Implementation

USB Descriptors

USB Registration

Initializing USB Interfaces

00 966 92000 9278

USB Drivers

training@globalknowledge.com.sa

Releasing a PCI Device

Managing PCI Resources

Accessing Configuration Space

Initializing DMA

Deferred Activity: Work Queues

Netlink Sockets and User Events

Registering File Operations
 File I/O Conventions

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
   Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tving it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
   IOCTL Conventions

Implementing IOCTLs

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

Lockless IOCTLs

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Releasing a PCI Device

Listing PCI Devices

USB Transfer Types

USB Descriptors

www.globalknowledge.com/en-sa/

Managing PCI Resources

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices

Linux USB Implementation

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Block Device Registration

The gendisk structure

Initialization Example

Linux Kernel Variants

Kernel Implementation

The Linux Source Tree

Writing Kernel Modules

Device Driver Design

Device Nodes

File Operations

Compiling Kernel Modules

More about Kernel Modules

Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: TaskletsDeferred Activity: Work Queues

Custom Work Queues

Example: Serial Mouse

IOCTL Conventions

Lockless IOCTLs

Implementing IOCTLs

00 966 92000 9278

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration
 System Context

Softirgs

training@globalknowledge.com.sa

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Interacting with the Kernel

Handling Requests

Request Queues

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Driver Implementation

Network Device Callback Functions

Network Driver Transmission Queue

- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures

The Device Node Problem

Defining File Operations

Device Model Structures

Userspace event agents

Miscellaneous Devices

Deferred Activity: Tasklets

Custom Work Queues

Deferred Activity: Work Queues

Netlink Sockets and User Events

File I/O Conventions

The sysfs filesystem

Tying it all Together

Device Registration

Handling Interrupts

Interrupt Structures

ISR Registration

System Context

Softirqs

www.globalknowledge.com/en-sa/

Registering File Operations

Interacting with Character Device Drivers

Interacting with Network Interface Cards

Character Device Drivers: Registration

Interacting with Block Device Drivers

Device Driver Design

Device Nodes

File Operations

- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests
- Linux Kernel Variants
- Interacting with the Kernel
- Kernel Implementation
- The Linux Source Tree
- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirqs

- Deferred Activity: Tasklets
- Deferred Activity: Work Queues

Example: Serial Mouse

Sleeping

Wait Queues

Wait Queues

Multiplexed I/O

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices

Listing PCI Devices

USB Transfer Types

USB Descriptors

USB Registration

USB Class Drivers

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Control

USB Request Blocks

USB Drivers

Releasing a PCI Device

Managing PCI Resources

Linux USB Implementation

Initializing USB Interfaces

Using USB request blocks

Synchronous USB Requests

Network Driver Initialization

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device Registration

The gendisk structure

Initialization Example

Handling Requests

training@globalknowledge.com.sa

Request Queues

Block Device File Operations

Block Device Driver Implementation

00 966 92000 9278

Network Device Callback Functions

Network Driver Transmission Queue

Accessing Configuration Space

Initializing DMA

Polling

I/O Ports

Variations on wait\_event()

Sleeping in System Calls

Managing Exclusive Access

Accessing I/O Memory Buffers

- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
   USB Class Drivers
- USB Request Blocks

More information

Network Devices

Network Statistics

Socket Buffer API

Transmission

I/O Scheduling

Reception

Socket Buffers (Packets)

Example Implementations

Linux 2.6 Elevator Functions

Block Device File Operations

Block Device Registration

The gendisk structure

Request Queues
 Initialization Example

Handling Requests

Linux Kernel Variants
 Interacting with the Kernel

Kernel Implementation

The Linux Source Tree

www.globalknowledge.com/en-sa/

Block Device Driver Implementation

Control

- Using USB request blocks
- Synchronous USB Requests

Network Driver Initialization

Network Device Callback Functions

Network Driver Transmission Queue

- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls
- Multiplexed I/O
- Polling
- Managing Exclusive Access
- I/O Ports
- I/O Memory Buffers
- Accessing I/O Memory Buffers
- Userspace Drivers
- Initializing DMA
- Coherent Allocations
- Streaming DMA Mappings
- PCI Configuration Space
- PCI Driver Registration
- Registering PCI Drivers
- Driver Probe Function
- Initializing PCI devices
- Releasing a PCI Device
- Managing PCI Resources
- Accessing Configuration Space
- Listing PCI Devices
- USB Descriptors
- USB Transfer Types
- Linux USB Implementation
- USB Drivers
- USB Registration
- Initializing USB Interfaces
- USB Class Drivers
- USB Request Blocks
- Using USB request blocks
- Synchronous USB Requests
- More information
- Network Devices
- Network Device Callback Functions
- Network Driver Initialization
- Network Driver Transmission Queue Control
- Network Statistics
- Socket Buffers (Packets)
- Socket Buffer API
- Transmission
- Reception
- Example Implementations
- I/O Scheduling
- Linux 2.6 Elevator Functions
- Block Device Driver Implementation
- Block Device Registration
- Block Device File Operations
- The gendisk structure
- Request Queues
- Initialization Example
- Handling Requests

- Writing Kernel Modules
- Compiling Kernel Modules
- More about Kernel Modules
- Review of Kernel Structures
- Device Driver Design
- Interacting with Character Device Drivers
- Interacting with Block Device Drivers
- Interacting with Network Interface Cards
- Device Nodes
- Character Device Drivers: Registration
- The Device Node Problem
- File Operations
- Defining File Operations
- Registering File Operations
- File I/O Conventions
- The sysfs filesystem
- Device Model Structures
- Tying it all Together
- Netlink Sockets and User Events
- Userspace event agents
- Device Registration
- Miscellaneous Devices
- Handling Interrupts
- Interrupt Structures
- ISR Registration
- System Context
- Softirgs
- Deferred Activity: Tasklets
- Deferred Activity: Work Queues
- Custom Work Queues
- Example: Serial Mouse
- IOCTL Conventions
- Implementing IOCTLs
- Lockless IOCTLs
- Sleeping
- Wait Queues
- Wait Queues
- Variations on wait\_event()
- Sleeping in System Calls

I/O Memory Buffers

Userspace Drivers

Coherent Allocations

Streaming DMA Mappings

PCI Configuration Space

PCI Driver Registration

Registering PCI Drivers

Driver Probe Function

Initializing PCI devices
 Releasing a PCI Device

Listing PCI DevicesUSB Descriptors

USB Transfer TypesLinux USB Implementation

USB Drivers
USB Registration
Initializing USB Interfaces
USB Class Drivers
USB Request Blocks

www.globalknowledge.com/en-sa/

Managing PCI Resources

Accessing Configuration Space

training@globalknowledge.com.sa

00 966 92000 9278

Initializing DMA

Accessing I/O Memory Buffers

- Multiplexed I/O
- Polling
   Managing Exclusive Access

I/O Ports

Using USB request blocks Synchronous USB Requests More information Network Devices Network Device Callback Functions Network Driver Initialization Network Driver Transmission Queue Control Network Statistics Socket Buffers (Packets) Socket Buffer API Transmission Reception Example Implementations I/O Scheduling Linux 2.6 Elevator Functions Block Device Driver Implementation Block Device Registration Block Device File Operations The gendisk structure Request Queues Initialization Example Handling Requests

# **Further Information:**

For More information, or to book your course, please call us on 00 966 92000 9278

training@globalknowledge.com.sa

www.globalknowledge.com/en-sa/

Global Knowledge - KSA, 393 Al-Uroubah Road, Al Worood, Riyadh 3140, Saudi Arabia