



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 4. 1. 1. Static
 4. 1. 1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.



2-Day Embedded Linux Development

Overview

This two day course provides experience with the key steps in developing an embedded Linux product.

The course begins with a complete, simple, project that walks the attendees through the entire process of creating a special version of the Linux kernel, creating a root file system, including just the libraries that are needed, and constructing a custom boot sequence.

Attendees will spend approximately 50 percent of the class time actually gaining hands-on experience with these topics.

Course Objectives

- To provide an understanding of the essentials of embedded Linux.
- To give you practical experience in developing an embedded Linux system.

Attendees will learn:

- Key concepts and software for embedded Linux.
- Fundamentals of embedded software for Linux

Who Should Attend:

The course is designed for real-time or embedded engineers who are new to real-time or embedded Linux. Attendees should have experience with C and be able to perform basic Unix commands.

Duration

Two days

Course Materials

The workshop materials include a comprehensive student workbook and CD.

The workbook contains all of the slides used in the course as well as hands-on lab exercises.

The CD contains the lab exercise code as well as a large amount of embedded Linux software.

Course Workshop:

The workshop makes use of standard PC's with a desktop Linux distribution for development. Platforms such as those with MIPS, ARM, or PPC CPUs will be used as for some target and cross-development portions of the course. Since Linux has been effectively ported to many architectures, the principles taught in the workshop are appropriate for a wide range of target platforms.



2-Day Embedded Linux Development Outline

- 1. Embedded Linux Development**
 - 1.1. Objectives and format
 - 1.2. What/Why/How/Who/Where of embedded Linux
- 2. Overview Of Project**
 - 2.1. Building an mp3 playing, web browser controlled, appliance
 - 2.2. Configuring a Linux kernel
- 3. Building A Root File System**
 - 3.1. What directories are required?
 - 3.2. Making Busybox
 - 3.3. Configuring the boot sequence
 - 3.4. Configuring networking
- 4. Building A System Image**
 - 4.1. Device drivers
 - 4.1.1. Static
 - 4.1.1. Dynamic
 - 4.2. Inserting drivers
 - 4.3. Stacked drivers
 - 4.4. Libraries
- 5. Applications**
 - 5.1. running an embedded web server
 - 5.2. application software
- 6. Making A Boot Image**
 - 6.1. Boot loaders
 - 6.2. Putting the pieces together
 - 6.3. Building a complete bootable system image
- 7. Cross development**
 - 7.1. Cross compilation
 - 7.2. Libraries and tool chains
 - 7.3. How to configure the kernel for cross-compiling
 - 7.4. Building the kernel and modules
 - 7.5. Creating applications on the target.