

Kernel Modification and compilation

Hemil Modi - 121016

Semester 2015-2

Why Compile Kernel?

There are two main reasons, why would one compile his/her own kernel.

- 1 Remove Code for which there is no hardware installed in the system. Example: Nvidia graphics card
- 2 If the your system has a special hardware and your latest version of Ubuntu does not come with it's driver than you may want to have that in your kernel. Example: Fingerprint reader

Why Compile Kernel?

There are two main reasons, why would one compile his/her own kernel.

- 1 Remove Code for which there is no hardware installed in the system. Example: Nvidia graphics card
- 2 If the your system has a special hardware and your latest version of Ubuntu does not come with it's driver than you may want to have that in your kernel. Example: Fingerprint reader

After Downloading and extracting a kernel, you will see many files and folder. There are some files in root source code:

- 1 Kbuild
- 2 Kconfig
- 3 MAINTAINERS
- 4 Makefile
- 5 README
- 6 init
- 7 include
- 8 kernel

After Downloading and extracting a kernel, you will see many files and folder. There are some files in root source code:

- 1 Kbuild
- 2 Kconfig
- 3 MAINTAINERS
- 4 Makefile
- 5 README
- 6 init
- 7 include
- 8 kernel

After Downloading and extracting a kernel, you will see many files and folder. There are some files in root source code:

- 1 Kbuild
- 2 Kconfig
- 3 MAINTAINERS
- 4 Makefile
- 5 README
- 6 init
- 7 include
- 8 kernel

After Downloading and extracting a kernel, you will see many files and folder. There are some files in root source code:

- 1 Kbuild
- 2 Kconfig
- 3 MAINTAINERS
- 4 Makefile
- 5 README
- 6 init
- 7 include
- 8 kernel

After Downloading and extracting a kernel, you will see many files and folder. There are some files in root source code:

- 1 Kbuild
- 2 Kconfig
- 3 MAINTAINERS
- 4 Makefile
- 5 README
- 6 init
- 7 include
- 8 kernel

After Downloading and extracting a kernel, you will see many files and folder. There are some files in root source code:

- 1 Kbuild
- 2 Kconfig
- 3 MAINTAINERS
- 4 Makefile
- 5 README
- 6 init
- 7 include
- 8 kernel

After Downloading and extracting a kernel, you will see many files and folder. There are some files in root source code:

- 1 Kbuild
- 2 Kconfig
- 3 MAINTAINERS
- 4 Makefile
- 5 README
- 6 init
- 7 include
- 8 kernel

After Downloading and extracting a kernel, you will see many files and folder. There are some files in root source code:

- 1 Kbuild
- 2 Kconfig
- 3 MAINTAINERS
- 4 Makefile
- 5 README
- 6 init
- 7 include
- 8 kernel

Configuring Your kernel

Few of the many ways to configure a kernel

- 1 make config
- 2 make menuconfig
- 3 make nconfig
- 4 make xconfig - QT required
- 5 make jconfig - GTK required
- 6 make deconfig

Configuring Your kernel

Few of the many ways to configure a kernel

- 1 make config
- 2 make menuconfig
- 3 make nconfig
- 4 make xconfig - QT required
- 5 make jconfig - GTK required
- 6 make deconfig

Configuring Your kernel

Few of the many ways to configure a kernel

- 1 make config
- 2 make menuconfig
- 3 make nconfig
- 4 make xconfig - QT required
- 5 make jconfig - GTK required
- 6 make deconfig

Configuring Your kernel

Few of the many ways to configure a kernel

- 1 make config
- 2 make menuconfig
- 3 make nconfig
- 4 make xconfig - QT required
- 5 make jconfig - GTK required
- 6 make deconfig

Configuring Your kernel

Few of the many ways to configure a kernel

- 1 make config
- 2 make menuconfig
- 3 make nconfig
- 4 make xconfig - QT required
- 5 make jconfig - GTK required
- 6 make deconfig

Configuring Your kernel

Few of the many ways to configure a kernel

- 1 make config
- 2 make menuconfig
- 3 make nconfig
- 4 make xconfig - QT required
- 5 make jconfig - GTK required
- 6 make deconfig

Configuring Your kernel

Few of the many ways to configure a kernel

- ❶ make config
- ❷ make menuconfig
- ❸ make nconfig
- ❹ make xconfig - QT required
- ❺ make jconfig - GTK required
- ❻ make deconfig

Compile Install linux kernel

C language to machine language.

Commands to compile a kernel:

- 1 make
- 2 make module

For ease of compiling process use:

make;make modules:

After compilation both kernel and modules can be compiled using:

“make install make modules;install”

“uname -r”: to ensure if the kernel is installed.

DevynCJohnson

<http://www.linux.org/threads/the-linux-kernel-series-every-article.6558/>