Linux Kernel Custom Configuration and Compilation

---

1. Introduction

The Linux kernel is the core of the Linux operating system. Customizing the kernel allows you to enable or disable features, add drivers, and optimize for specific hardware.

The .config file in the kernel source directory stores all configuration options.

---

2. Kernel Source

Installing prerequisites (Ubuntu/Debian):

```
sudo apt update
sudo apt install build-essential libncurses-dev bison flex libssl-dev libelf-dev git
```

Download Linux kernel source:

```
git clone https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git
cd linux
git checkout v6.6
```

---

3. Understanding .config

- .config defines all kernel options
- Options can be built-in [Y], modular [M], or disabled [ ].
- Location: kernel source root (linux/.config).

Example entries:

```
CONFIG_USB_SUPPORT=y
CONFIG_USB_SERIAL=m
CONFIG_NETFILTER=y
```

---

4. Kernel Configuration Methods

A) Menu-driven configuration (text-based)

```
make menuconfig
```

- Navigate with arrows.

- Y = build-in, M = module, N = disable.

B) Graphical configuration

```
make xconfig   # Qt-based GUI
make gconfig   # GTK-based GUI
```

C) Copy existing config

```
cp /boot/config-$(uname -r) .config
make oldconfig
```

- Updates .config from existing kernel.

D) Other CLI options

```
make nconfig    # Improved CLI with descriptions
make olddefconfig  # Accept defaults for new options
```

---

5. Important Kernel Configuration Sections

1. Processor type and features
    - CPU architecture, SMP support, virtualization.
2. Device Drivers
    - USB, SPI, I2C, GPIO, network cards.
3. File systems
    - EXT4, FAT, NTFS, NFS.
4. Networking Support
    - TCP/IP, Bluetooth, Wi-Fi.
5. Kernel Hacking
    - Debugging, printk levels, memory debugging.

Tip: Read the help (?) for each option to understand dependencies.

---

6. Compiling the Kernel

```
# Compile kernel with all cores
make -j$(nproc)

# Install modules
```

```
sudo make modules_install

# Install kernel
sudo make install
```

- Installs kernel to /boot.
- Updates GRUB automatically.

6.1 Reboot and verify

```
sudo reboot
uname -r  # Shows current running kernel
```

---

7. Example: Enabling USB UART Driver

1. Run:

```
make menuconfig
```

2. Navigate: Device Drivers -> Character devices -> USB Serial Converter
3. Select [M] for module.
4. Compile:

```
make -j$(nproc)
make modules_install
make install
```

5. Load module:

```
sudo modprobe usbserial
```

---

8. Tips for Custom Kernel

- Backup .config:

```
cp .config ~/kernel-config-backup
```

- Enable features as modules first to avoid boot failures.

- Check dependencies for drivers.
- Use make nconfig for a detailed CLI interface.
- Read the kernel documentation in Documentation/ folder for each subsystem.

---

9. References

1. Kernel Newbies
2. Linux kernel source Documentation/ directory
3. make help inside kernel source directory
