ChatGPT

Table of Contents

how to install linux 6.12 only kernel and user, file system, syetem

To install and run a specific Linux kernel version like 6.12 with a minimal userland and filesystem, follow these steps. This process is generally for advanced users or developers who want a lightweight or customized Linux environment.

Prerequisites

1. **Basic Tools**: You need tools like `gcc`, `make`, and other development libraries. You can install these with:

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

2. **Kernel Source**: Download the source code for Linux kernel 6.12 from kernel.org.

3. **Disk Space**: Ensure sufficient space for kernel compilation and the minimal system.

Step 1: Download and Build the Linux Kernel

1. **Download the kernel**:

```bash
wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.12.tar.xz
tar -xvf linux-6.12.tar.xz
cd linux-6.12
```

2. **Configure the kernel**:

```bash
make menuconfig
```

   - Include only the necessary drivers and features.
   - Ensure support for the minimal filesystem you plan to use (e.g., ext4).

3. **Build the kernel**:

```bash
make -j$(nproc)
make modules
sudo make modules_install
sudo make install
```

4. **Update Bootloader**: If using GRUB, update the configuration:

```bash
sudo update-grub
```

   Reboot and select the new kernel from the GRUB menu.

Step 2: Create a Minimal Root Filesystem

You need a minimal set of binaries and libraries for a functional userland:

1. **Use** `busybox`: It provides essential utilities in a compact form.
   - Download and compile `busybox`:

```bash
wget https://busybox.net/downloads/busybox-<version>.tar.bz2
tar -xvf busybox-<version>.tar.bz2
cd busybox-<version>
make menuconfig
```

```
make
sudo make install
```

- Install `busybox` to `/bin`.

2. **Set Up Directory Structure**: Create a directory tree for your minimal filesystem:

```bash
mkdir -p rootfs/{bin,dev,etc,lib,proc,sys}
cp -a /usr/bin/busybox rootfs/bin/
```

3. **Create Basic Configuration Files**:
   - Add `/etc/inittab` for the initialization process.
   - Configure `fstab` for mounting filesystems.
   - Create necessary device files under `/dev`.
4. **Build an Initramfs**: Use the `find` command to generate an initial ramdisk:

```bash
cd rootfs
find . | cpio -o -H newc | gzip > ../initramfs.gz
```

Step 3: Boot the Kernel with the Filesystem

1. **Update the Bootloader**: Edit the GRUB configuration to load your custom `initramfs`:

```bash
sudo nano /boot/grub/grub.cfg
```

   Add the kernel and `initramfs` entries.

2. **Reboot the System**: Select your custom kernel and filesystem from GRUB.

Troubleshooting

- If the system fails to boot, check logs by adding the `debug` option to the kernel command line in GRUB.
- Use QEMU or a virtual machine for testing before deploying on physical hardware.

Would you like additional details or guidance on a specific part of the process?

I will build on harddisk with using ubuntu live cd

To build and install Linux Kernel 6.12 directly onto a hard disk using an Ubuntu Live CD, you can follow this approach. This assumes you are using the Ubuntu Live CD environment for a minimal system setup and kernel building. Here's the detailed process:

Prerequisites

1. **Ubuntu Live CD**: Boot from an Ubuntu Live CD or USB.
2. **Internet Connection**: Ensure network access for downloading tools and kernel sources.
3. **Partition**: Prepare a dedicated partition or hard disk to install the kernel and minimal system.

Step 1: Prepare the Environment

1. **Boot the Ubuntu Live CD**:
   - Select "Try Ubuntu" to enter the live environment.
2. **Update and Install Build Tools**: Open a terminal and run:

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

Step 2: Download and Build the Kernel

1. **Download the Kernel Source**:

```bash
wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.12.tar.xz
tar -xvf linux-6.12.tar.xz
cd linux-6.12
```

2. **Configure the Kernel**:

```bash
make menuconfig
```

   - Select the necessary drivers for your hardware.
   - Add support for the minimal filesystem (e.g., ext4, FAT).
3. **Compile the Kernel**:

```bash
```

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

Step 3: Create a Minimal Root Filesystem

1. **Format and Mount the Target Partition**:
   - Identify the target partition (e.g., `/dev/sdX1`) using `lsblk`.
   - Format the partition:
     ```bash
     sudo mkfs.ext4 /dev/sdX1
     ```
   - Mount it:
     ```bash
     sudo mount /dev/sdX1 /mnt
     ```
2. **Install BusyBox**:
   - Download and build `busybox`:
     ```bash
     wget https://busybox.net/downloads/busybox-<version>.tar.bz2
     tar -xvf busybox-<version>.tar.bz2
     cd busybox-<version>
     make menuconfig
     make
     sudo make install
     ```
   - Copy `busybox` to the mounted partition:
     ```bash
     sudo cp -a /usr/bin/busybox /mnt/bin/
     ```
3. **Set Up Directory Structure**: Create the necessary directories on the mounted partition:
   ```bash
   sudo mkdir -p /mnt/{bin,dev,etc,proc,sys}
   ```
4. **Copy Kernel and Initramfs**: Copy the compiled kernel and `initramfs` to the `/boot` directory of the target partition:
   ```bash
   sudo cp /boot/vmlinuz-6.12.* /mnt/boot/
   ```

```
sudo cp initramfs.gz /mnt/boot/
```

5. **Create Basic Configuration Files**:
   - Add an `inittab` file to `/etc`.
   - Set up an `fstab` for the minimal filesystem.
   - Populate `/dev` with essential device files.

Step 4: Install a Bootloader

1. **Install GRUB on the Target Disk**:
   - Mount necessary directories:
     ```bash
     sudo mount --bind /dev /mnt/dev
     sudo mount --bind /proc /mnt/proc
     sudo mount --bind /sys /mnt/sys
     ```
   - Chroot into the mounted partition:
     ```bash
     sudo chroot /mnt
     ```
   - Install GRUB:
     ```bash
     grub-install /dev/sdX
     update-grub
     exit
     ```
2. **Edit GRUB Configuration**: Ensure the kernel and `initramfs` paths are correct in `/boot/grub/grub.cfg`.

Step 5: Test the Installation

1. **Reboot**: Reboot your system and select the new kernel from GRUB.
2. **Verify**: Once booted, confirm the kernel version with:
   ```bash
   uname -r
   ```

Would you like detailed help with any of these steps, or do you need guidance on troubleshooting?