

Build Linux Kernel From Scratch

Creating the Build Environment

The Differences Between Kernels, Operating Systems, and Distributions

- The kernel is the brain of Linux, and the core interface between the computer's hardware and its processes
- Kernel has 4 jobs: Memory management, process management, device drivers, and system calls & security
- An OS includes the kernel and then, also, system services, applications and user accounts
- A distribution is a set of specific software packages added to the kernel
- Linux is all three of these

Preparing For the Build

- Planning stage: gather info needed for the build
- Need a system running Linux to build on, 12 gb of storage, access to the command line, and a user account with sudo/root privileges
- Need to download the latest stable kernel source code and verify its integrity

Setting Up Our Virtual Machine

- Platforms to build a Linux kernel: Standalone Linux server, or virtual.

Building the Kernel

Downloading & Extracting the Source Code

- Download from kernel.org
- Best practice is to download one release before the latest release for stability and bug fixes
- `wget https://cdn.kernel.org/pub/linux/kernel/v6.x`
- `tar xvf linux-6.0.1.tar.xz`

Installing the Required Packages

- Required: git, fakeroot, build-essential, ncurses-dev, xz-utils, libssl-dev, bc, flex, libelf-dev, bison

Configuring & Building the Kernel

- Change directory into the Linux kernel source code
- Copy the default configuration file into that directory location

- Open the config file for editing, edit the CONFIG_SYSTEM_TRUSTED_KEYS directive
- `make menuconfig`
- Select the defaults now
- `make`
- `make install`
- Copy config file: `cp -v /boot config-$(uname -r)*`
- If SSH Timeout, add & at end of command to run in the background, or edit client timeout interval in SSH config

Updating the Bootloader

- `sudo update-grub`

Verifying the Environment

- Reboot the server
- `uname -mrs`

Customizing the Kernel

Optimizing the Kernel for the Cloud

- There are a few reasons why we might do this:
 - To reduce performance lag
 - less resource intensive
- These settings must be configured during `make menuconfig`
- Uncheck the prompt for Development and/or Incomplete Code Drivers
- Uncheck CPU Set Support unless you have more than 1 processor (multicore processors count as more than 1)
- In the block layer, uncheck everything unless you have disks that are larger in size than 2 terabytes
- At the Processor Type and Features, uncheck Symmetric Multi-Processing Support unless you have more than 1 processor
- Uncheck everything under Kernel Hacking
- To save memory:
 - In Networking, uncheck Amateur Radio Support
 - Under Filesystems and Partition Types, uncheck anything that isn't valid
 - In device drivers, we're going to compare our devices. Most can be removed
 - Check manufacturer specifications for hardware in your cloud
 - You can disable EISA and MCA support unless your motherboards use these buses

Creating a Linux Distribution

- A distro consists of hardware, the kernel space, and it's an operating system.
- Since it's an operating system, it should include the user space, which includes user programs, application programs and libraries, and system programs
- Download the latest stable release of [Linux From Scratch](#)