Log Book Week 12

Jose Devian Hibono
1706039603
System Programming - C

This week’s material is about Boot Sequence and Kernel Compilation. First I watched the video and found some resources on the internet. Here are the resources I gathered, I tried to write my own understandings in this log book.

Boot Sequence

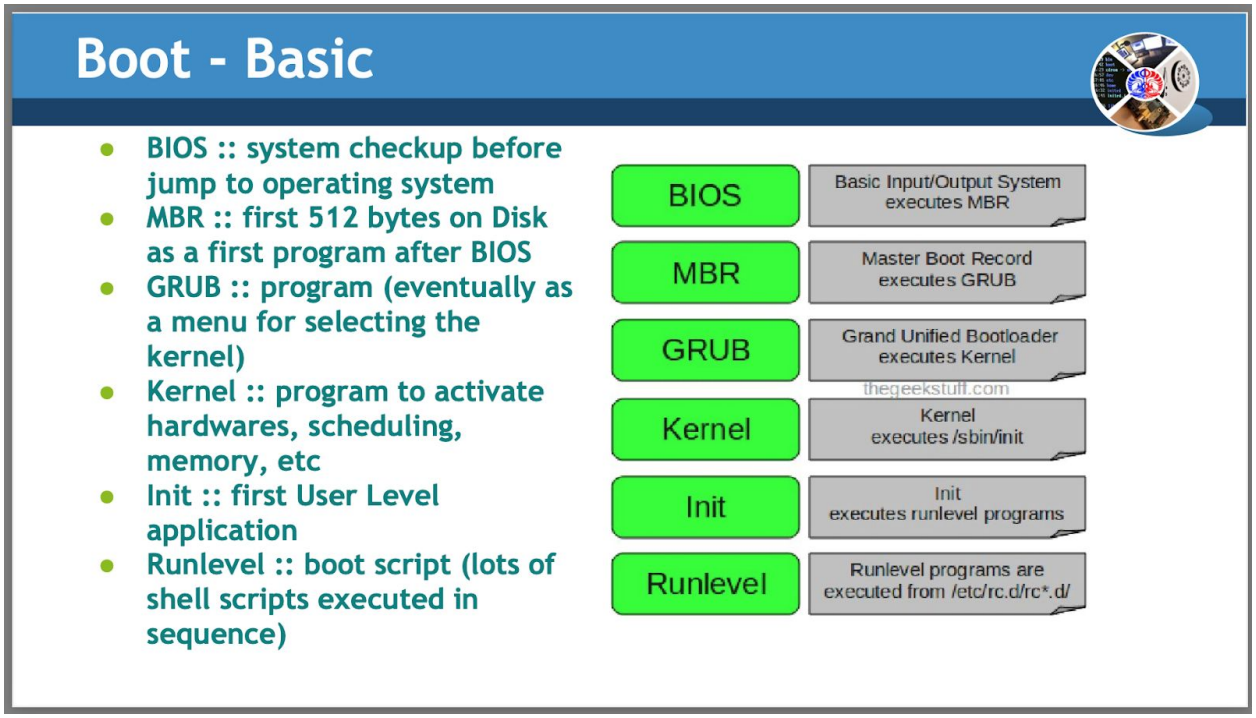
Boot Loader

A boot loader is a type of program that loads and starts an operating system or the computer system's boot time tasks and processes. It requires the operating system to be loaded into the computer's memory when a computer begins or boots up.

A boot loader is also referred to as a boot manager or loader for bootstraps.

A boot loader primarily controls and executes a computer system's boot sequence. After the computer or BIOS has finished performing the initial power and hardware system checks and tests, a boot loader program is usually started. Retrieves the OS kernel to the main memory from the hard drive or any designated boot device within the boot series. Only a single operating system is connected with a boot loader.

Multiple boot loader programs can also be categorized as primary and secondary boot loaders by an operating system, where a secondary boot loader may be larger and more powerful than the primary boot loader.



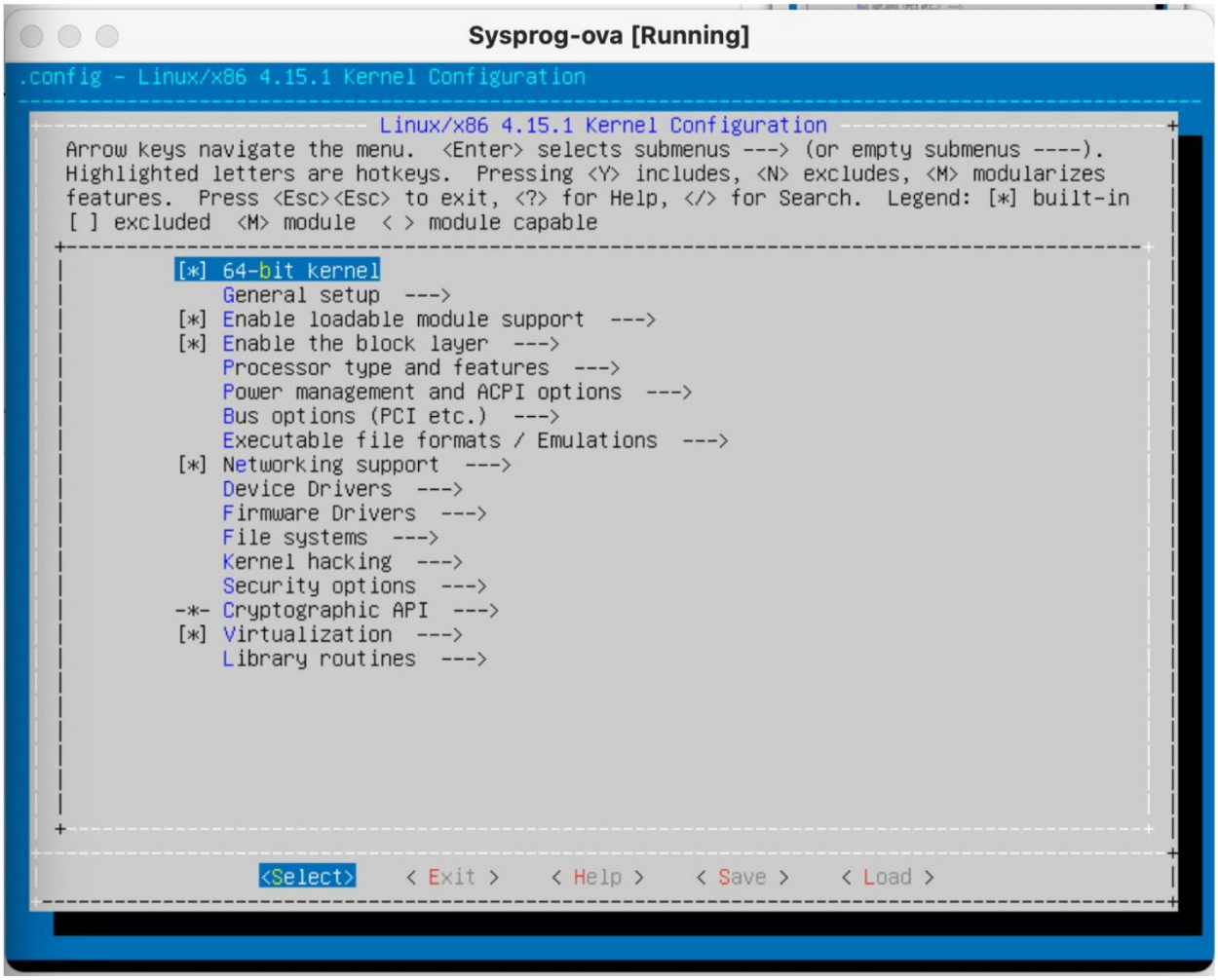
Taken from the course's slides.

Systemd

`systemd` is a suite of basic building blocks for a system running Linux. It provides a system and service manager that starts the rest of the system and runs as PID 1. `systemd` provides aggressive parallelization features, uses start-up services socket and D-Bus activation, offers daemon start-up on-demand, keeps track of processes using Linux control groups, manages mount and automount points, and implements an elaborate logic for service control based on transactional dependence. `systemd` supports init scripts for SysV and LSB and acts as a substitute for `sysvinit`.

Kernel Compilation

menuconfig



We need to compile our kernel to make the system stable with its hardware.

Kernel Architectural Approach

Monolithic

A monolithic kernel is the architecture of the operating system where the whole operating system runs in kernel space. The monolithic paradigm varies in that it alone describes a high-level virtual interface over computer hardware from other operating system architectures (such as the microkernel architecture).

Micro-kernel

A microkernel is a minimal kernel of a computer operating system that offers no operating system services at all in its purest form, only the mechanisms required to enforce those services, such as low-level space management of addresses, thread management, and inter-process communication (IPC).

Hybrid

The hybrid kernel is a kernel architecture used in computer operating systems based on a combination of a microkernel and a monolithic kernel architecture. This approach to the kernel combines the speed and simplified nature of the monolithic kernel with microkernel modularity and execution protection.

In order to minimize the overhead efficiency of a conventional microkernel, a hybrid kernel runs some services in the kernel space, while still running kernel code as servers in the user space. For example, as user mode programs outside the kernel, a hybrid kernel design may carry the virtual file system and bus controllers within the kernel and the file system drivers and storage drivers. The efficiency and design concepts of a monolithic kernel are retained by such a design.

A well-known example of a hybrid kernel powered by Windows NT, Windows 2000, Windows XP, Windows Server 2003, Windows Vista, Windows Server 2008 and Windows 7 is the Microsoft NT kernel. As emulation subsystems run in the user mode server processes, it is referred to as a monolithic kernel.

The structure, which is a set of modules that communicate through well-known interfaces, is one of the most important things about it, with a small microkernel restricted to core functions such as first-level interrupt handling, thread scheduling and primitive synchronization. This makes it possible to use either direct procedure calls or interprocess communication to communicate between modules, thereby allowing modules to theoretically be put in different address spaces.

References

Techopedia. (n.d). *Hybrid Kernel*. Accessed Dec 8, 2020 from https://www.techopedia.com/definition/27004/hybrid-kernel.

Techopedia. (n.d). *Boot Loader*. Accessed Dec 8, 2020 from https://www.techopedia.com/definition/3324/boot-loader.

Faculty of Computer Science, Universitas Indonesia. (n.d.). *Lecture 13: Boot Sequence & Kernel Compilation*.