Linux Build/Configuration

Deepak Ravi

Overview

- Linux
  - Lab: Download linux kernel source code
  - Lab: Install dependency packages
  - Lab: do make menuconfig
  - Lab: do make

Linux Kernel Build

- Step 1: Edit/Generate .config
- Step 2: make
- Step 3: ?

Why do we need OS?

- If there is only one trusted application ever running..
- If the applications are all trusted..

Linux kernel as a secure multiplexer of resources

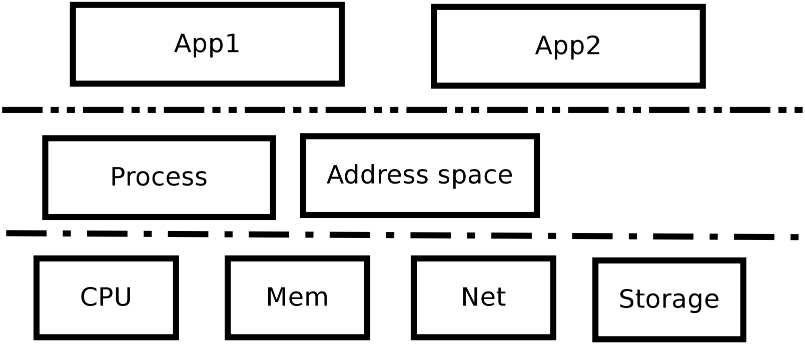

Figure 1:

- Map infinte resources to finite resources
- Isolation/Controlled sharing

Linux Kernel Build

- Step 0: Get the source code (git clone)
- Step 1: Switch to right branch/tag. (git checkout)
- Step 2: Edit/Generate .config
- Step 3: make
- Step 4: ?

Linux source code repository

- Stable: https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git
- Main: https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git

Can we have an UI for generating .config

- Need to know the type of each parameter:
  - Bool : Is it an Y or N
  - Tristate: Is it Y or N or M
  - Int : an integer value
  - String..
- How to get these information?

Can we have an UI for generating .config

- Yes, Linux Kernel's solution:
  - Specify type of each parameter in Kconfig file
    config MY_READ bool "Enable read system call" default y
  - For different types of UI:
    - make config : cli
    - make menuconfig : tui
    - make xconfig : gui

Demo

- Step 0: Get the source code (git clone)
- Step 1: Switch to right branch/tag (git checkout)
- Step 2: Edit/Generate .config (make menuconfig)
- Step 3: make
- Step 4: ?

Demo

- How to modify the configuration options?
- How to add a new configuration option for your new driver?
- What happens when you modify the option?
- How does Makefile uses this generated option?
- Can your C code driver uses this config information?

Need for kernel modules

- What's the issue with all the drivers compiled into single executable/kernel?

Need for kernel modules

- What's the issue with all the drivers compiled into single executable/kernel?
    - Increase in size. Not all drivers may not be needed by everyone
    - Updating a driver requires kernel update. Increase in devel time

Build steps

- Step 0: Get the source code (git clone)
- Step 1: Switch to right branch/tag (git checkout)
- Step 2: Edit/Generate .config (make menuconfig)
- Step 3: make
- Step 4: make modules
- But how to boot this kernel?

Build steps

- Step 0: Get the source code (git clone)
- Step 1: Switch to right branch/tag (git checkout)
- Step 2: Edit/Generate .config (make menuconfig)
- Step 3: make
- Step 4: make modules
- Step 5: sudo make install
- Step 6: sudo make modules_install
- But how to boot this kernel?

On Boot

- CPU sets cs:ip to 0xffff:0x0000 and starts executing code.
- CPU starts executing BIOS code directly from ROM.
- BIOS code initializes cache, RAM and other peripherals
- BIOS code installs its handlers IDT to provide services for bootloader
- BIOS loads the boot loader(grub2) code from the boot disk at 0x0000:0x7c00 and jump to it.
- Now, CPU starts executing boot loader code(grub2).
- (specific to grub2): grub2 uses bios provided interrupt handlers to load it's configuration file /boot/grub/grub2.cfg and gets the path of kernel to be loaded

Can we compile, load, unload a module without kernel source code?

- Need the header files (for your driver code)
  - sudo apt install linux-headers-amd64
- Need the linux kernel's Makefile, and the .config
- cd linux_headers_dir
- make modules M=path_to_your_external_module

Let's write a new kernel driver

- module_init(my_init): specify the function to be called on loading
- module_exit(my_exit): specify the function to be called on unloading

Let's write a new kernel driver

```
#include <linux/module.h>
MODULE_LICENSE("GPL");

static void my_exit(void){
  printk("Bye");
}
static void my_init(void){
  printk("Hello");
}

module_init(my_init);
module_exit(my_exit);
```

- But how to load/unload this driver?
- Why printk? Why not printf?
- Where to see the output of printk?