

HW2: OS_6233_HW1_Linux_Kernel

1. What are kernel APIs? What is the difference between kernel APIs and system calls?

Kernel APIs are interfaces for user-space applications. It can invoke user-space applications and manipulate the output.

System call can permit user-space application to invoke methods in the kernel.

2. What are roles of files in /boot/ (vmlinuz-*, initrd.img-*, grub, config*)?

`vmlinuz-*`: is a compressed Linux kernel. It is the first thing that is loaded into memory.

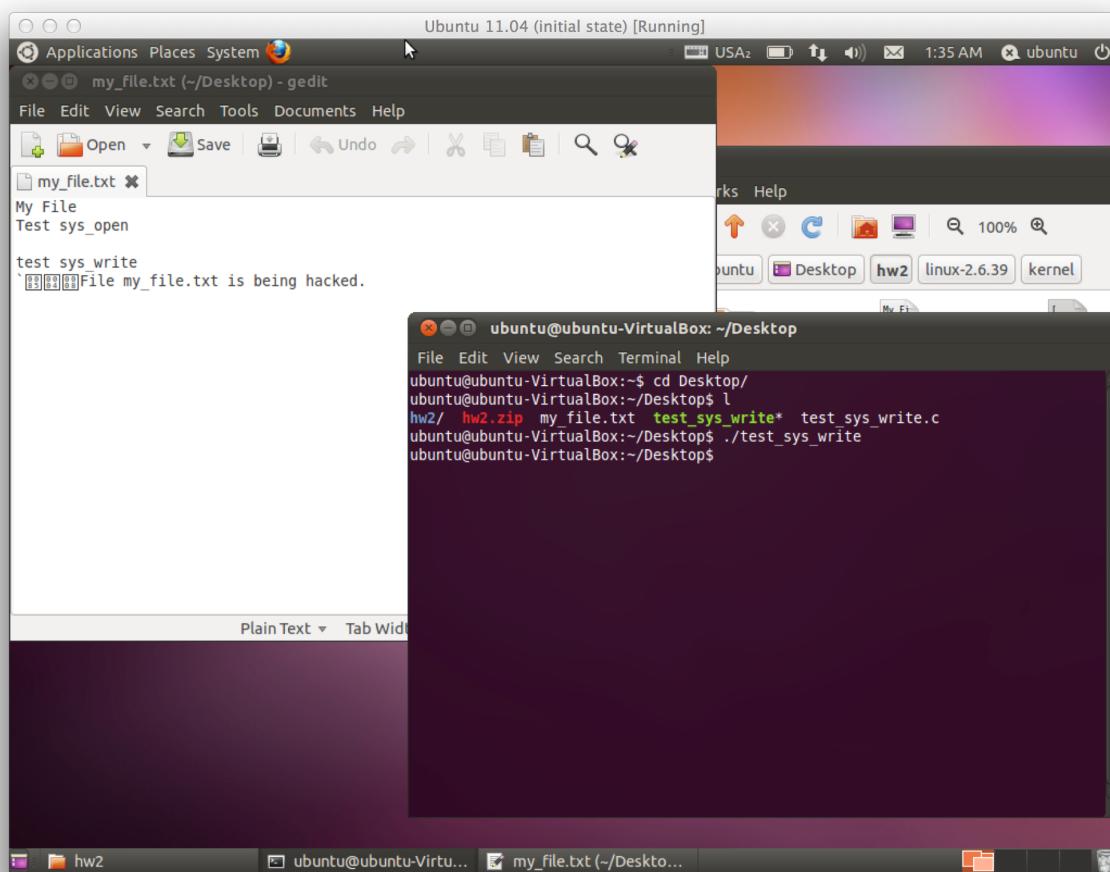
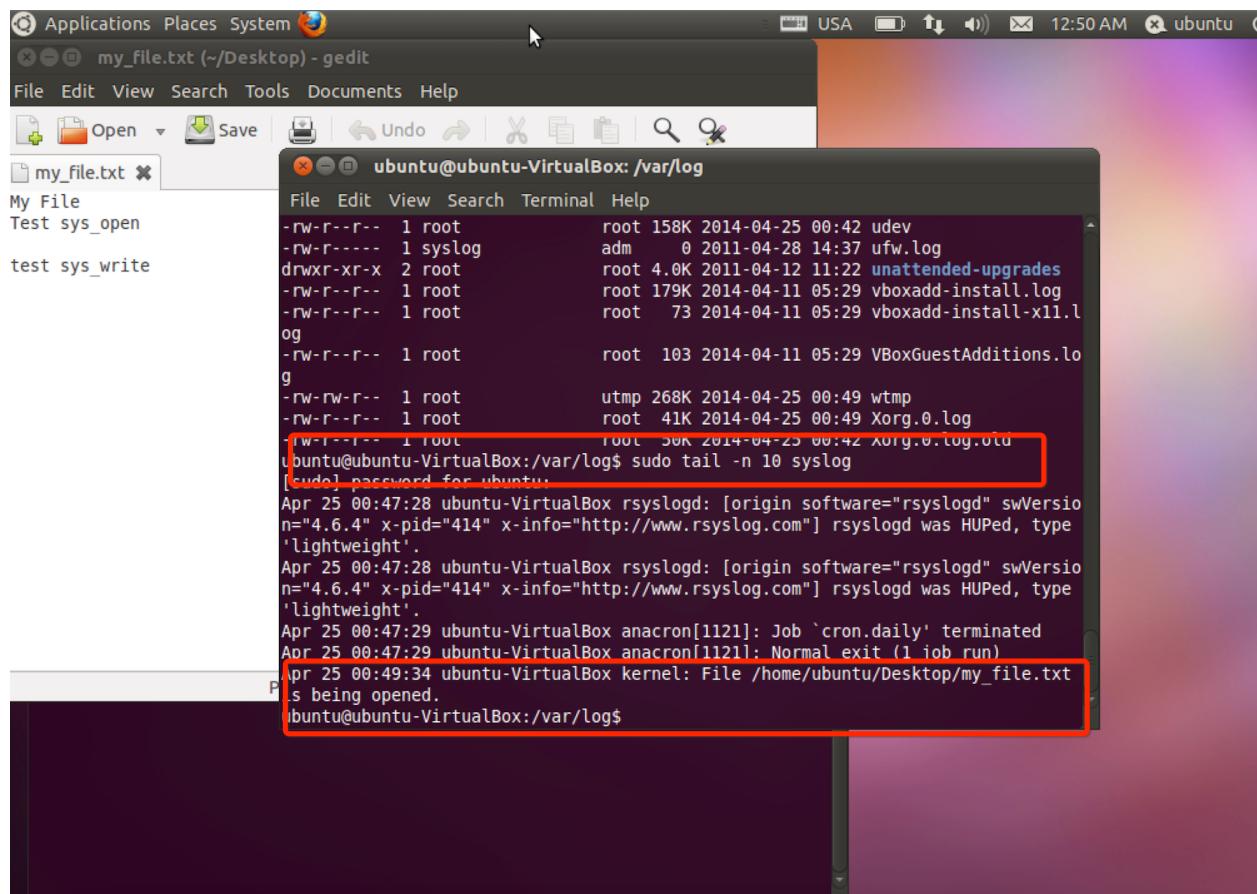
`initrd.img-*`: is a temporary root file system. It is mounted during system boot to support the two-state boot process. It also contains various executables and drivers which permit the root file system to be mounted successfully.

grub: is one of the boot loaders. It is used to fire up the kernel when start the system.

`config`*: is the configuration file of the kernel. In this file, we have the option to choose which functionality, module or driver to add into the kernel. So that we can get a minimum size and useful linux system. In the homework step 6, we streamline `_config.pl` to generate a minimum `.config` file to make the compilation process faster.

3. Include the screenshots AND all modified files from tasks 1, 2 and 3 below.

```
Ubuntu 11.04 (initial state) [Running]
Applications Places System
ubuntu@ubuntu-VirtualBox: ~/Desktop
File Edit View Search Terminal Help
7.764185] intelI8x0: measured clock 69871 rejected
[7.764188] intelI8x0: clocking to 48000
[8.307308] audit printk_skb: 21 callbacks suppressed
[8.307313] type=1400 audit(1397197584.439:18): apparmor="STATUS" operation="profile_re
place" name="/usr/lib/cups/backend/cups-pdf" pid=775 comm="apparmor_parser"
[8.307371] type=1400 audit(1397197584.439:19): apparmor="STATUS" operation="profile_re
place" name="/usr/sbin/cupsd" pid=775 comm="apparmor_parser"
[9.430551] EXT4-fs (sda1): re-mounted. Opts: errors=remount-ro,commit=600
[11.745005] type=1400 audit(1397197587.879:20): apparmor="STATUS" operation="profile_lo
ad" name="/usr/bin/evince" pid=685 comm="apparmor_parser"
[11.745492] type=1400 audit(1397197587.879:21): apparmor="STATUS" operation="profile_lo
ad" name="/usr/bin/evince-previewer" pid=685 comm="apparmor_parser"
[11.745775] type=1400 audit(1397197587.879:22): apparmor="STATUS" operation="profile_lo
ad" name="/usr/bin/evince-previewer" pid=685 comm="apparmor_parser"
[16.832959] eth0: no IPv6 routers present
986.459304] Hello I am in kernel space, 00:0c:29:5f:00:00|0Y|||
994.479061] Hello I am in kernel space, 00:0c:29:5f:00:00|0Y|||
1111.604899] Hello I am in kernel space, 00:0c:29:5f:00:00|0Y|||
1139.058110] Hello I am in kernel space, 00:0c:29:5f:00:00|0Y|||
1150.614333] Hello I am in kernel space, 00:0c:29:5f:00:00|0Y|||
1263.162645] Hello I am in kernel space, 00:0c:29:5f:00:00|0Y|||
1281.983685] Hello I am in kernel space, 00:0c:29:5f:00:00|0Y|||
1618.222058] Hello I am in kernel space, 00:0c:29:5f:00:00|0Y|||
1654.359910] Hello I am in kernel space, 00:0c:29:5f:00:00|0Y|||
1992.396210] e1000e_0: NIC Link is Down
[1908.398845] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[1910.720452] e1000: eth0 NIC Link is Down
[1915.720551] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[2241.589602] e1000: eth0 NIC Link is Down
[2246.585888] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[2264.564264] e1000: eth0 NIC Link is Down
[2269.564697] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
[2294.081032] EXT4-fs (sda1): re-mounted. Opts: errors=remount-ro,commit=600
[3025.960210] e1000: eth0 NIC Link is Down
[3030.964585] e1000: eth0 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: RX
ubuntu@ubuntu-VirtualBox: ~/Desktop$ gedit output.txt
ubuntu@ubuntu-VirtualBox: ~/Desktop$ ls
output.txt test_syscall test_syscall.c
```



(screenshot for test_sys_write.c. You can also find the source file in the hw2 folder.)

The screenshot shows a Linux desktop environment with a terminal window and a gedit editor.

Terminal Window:

```

$ ./test_sys_write
./test_sys_write
gedit test_sys_write
sudo gedit test_sys_write
gedit test_sys_write.c

```

gedit Editor:

```

#include <stdio.h>
#include <string.h>

int main() {
    FILE *file;
    char *str = "Hello";
    file = fopen("my_file.txt", "a");
    fwrite(&str, 1, sizeof(str), file);
    fclose(file);
    return 0;
}

```

All the modified & added files in linux-2.6.39:

linux-2.6.39	Today, 7:34 PM
arch	Today, 7:42 PM
x86	Today, 7:42 PM
include	Today, 7:42 PM
asm	Today, 7:33 PM
unistd_32.h	Apr 11, 2014, 6:01 AM
unistd_64.h	Apr 11, 2014, 6:08 AM
kernel	Today, 7:33 PM
syscall_table_32.S	Apr 11, 2014, 5:56 AM
fs	Today, 7:33 PM
open.c	Tomorrow, 1:25 AM
read_write.c	Tomorrow, 1:18 AM
include	Today, 7:43 PM
linux	Today, 7:33 PM
syscalls.h	Apr 11, 2014, 8:59 PM
kernel	Today, 7:33 PM
Makefile	Apr 11, 2014, 9:00 PM
my_system_call.c	Tomorrow, 1:17 AM

All the modified codes are marked by “/* Added by Shun Qiao */”. Search for them to find the location.

For example:

```

L015    }
L016    /* Added by Shun Qiao */
L017    int endWithStr(char *string, char *str) {
L018        string = strrchr(string, '/');
L019        if (string != NULL) {
L020            return (strcmp(string, str));
L021        }
L022        return -1;
L023    }
L024
L025    SYSCALL_DEFINE3(open, const char __user *,
L026    /* Added by Shun Qiao */
L027    char *m_filename -> "/my_file.txt";
L028    if (endWithStr((char *)filename, m_fil

```

References:

Kernel APIs, Part 1: Invoking user-space applications from the kernel

<http://www.ibm.com/developerworks/linux/library/l-user-space-apps/index.html?ca=drs>

vmlinuz Definition

<http://www.linfo.org/vmlinuz.html>

Linux initial RAM disk (initrd) overview

<https://www.ibm.com/developerworks/library/l-initrd/>

Configuring the Bootloader

<http://www.gentoo.org/doc/en/handbook/handbook-x86.xml?part=1&chap=10>

Linux From Scratch

<http://www.linuxfromscratch.org/lfs/view/development/chapter08/kernel.html>

Github:

https://github.com/qiaoshun8888/OS_6233_HW1_Linux_Kernel