

CS342 Operating Systems

Spring 2023
Homework #1

Efe Beydoğan
21901548
Section 2
CS

16/02/2023

Table of Contents

Installation.....3

Linux Commands.....3

Kernel.....4

C Program.....7

Makefile.....9

Installation

I opted to dual boot Ubuntu on my Windows laptop. In order to do so, I first had to allocate storage on my HDD for Ubuntu. I did this via the disk manager on Windows, and partitioned 200 GB storage. Then, I took a backup of sensitive files found on my hard drive to make sure I didn't lose any data in the installation process. After this, I downloaded Ubuntu 22.04 LTS from the given website on the homework document and also downloaded Rufus, a software that is used for preparing a bootable USB stick. I loaded the ISO image of Ubuntu onto the stick and using the BIOS screen of my laptop, I launched the setup application of Ubuntu. I made sure to choose the right partition to download Ubuntu into, and made the necessary settings, finally downloading and dual booting Ubuntu successfully. Finally, I downloaded Visual Studio Code from the Ubuntu Software store as that is the IDE I will be using for coding in C. Overall, the Ubuntu installation and dual booting process was without any issues and went smoothly.

Linux Commands

I found 10 commands via online tutorials on basic Linux usage. These commands and their explanations can be seen below:

1) pwd: The command name stands for “print working directory”. This command gives the absolute path (the path that starts from the root) of the working directory. The root is the base of the Linux system and is denoted by a “/”.

2) ls: The command name stands for “list”. This command is used for listing all the files in the current directory.

3) cd: The command name stands for “change directory”. This command is used for changing the current working directory like this: “*cd directoryName*”. It is case-sensitive.

4) mkdir: The command name stands for “make directory”. This command is used for creating a new folder like this: “*mkdir newDirectoryName*”.

5) rmdir: The command name stands for “remove directory”. This command is used to delete a directory like this: “*rmdir directoryName*”.

6) touch: This command is used to create a new file, such as an empty txt or a zip file. It can be used like this: “*touch file.txt*”.

7) cat: The command name stands for “concatenate”. This command is used to display the contents of one or more files without having to open the files, like this: “*cat file.txt*”.

8) clear: This command is used to clear the previous outputs in the terminal.

9) zip: This command is used to compress files into a zip archive.

10) unzip: This command is used to extract files from a zip archive.

Kernel

Location:

I found from the internet that the kernel executable is usually found in the `/boot` directory. I opened that directory and used the “`uname -r`” command, which output “5.15.0-58-generic”. The kernel executable is named `vmlinuz` and resides in the `/boot` directory.

```
efe@efe-ASUS-TUF-Gaming-F15-FX506LU-FX506LU:~$ cd ~/
efe@efe-ASUS-TUF-Gaming-F15-FX506LU-FX506LU:~$ cd ..
efe@efe-ASUS-TUF-Gaming-F15-FX506LU-FX506LU:/home$ cd ..
efe@efe-ASUS-TUF-Gaming-F15-FX506LU-FX506LU:/ $ cd boot
efe@efe-ASUS-TUF-Gaming-F15-FX506LU-FX506LU:/boot$ ls
config-5.15.0-43-generic      memtest86+.elf
config-5.15.0-58-generic      memtest86+_multiboot.bin
efi                           System.map-5.15.0-43-generic
grub                          System.map-5.15.0-58-generic
initrd.img                   vmlinuz
initrd.img-5.15.0-43-generic  vmlinuz-5.15.0-43-generic
initrd.img-5.15.0-58-generic  vmlinuz-5.15.0-58-generic
initrd.img.old               vmlinuz.old
memtest86+.bin
efe@efe-ASUS-TUF-Gaming-F15-FX506LU-FX506LU:/boot$ uname -r
5.15.0-58-generic
```

Figure 1: Kernel executable location

Version of the running kernel is 5.15.0-58-generic.

Kernel version 5.15.91 source code subdirectories:

```
efe@efe-ASUS-TUF-Gaming-F15-FX506LU-FX506LU:~/Desktop/linux-5.15.91$ ls
arch      CREDITS    fs          ipc         lib          mm          scripts    usr
block     crypto     include     Kbuild     LICENSES     net         security   virt
certs     Documenta init        Kconfig    MAINTAINERS README     sound
COPYING   drivers    io_uring    kernel     Makefile     samples    tools
```

Figure 2: Kernel version 5.15.91 subdirectories

System Call Table:

Location of the system call table: `~/Desktop/linux-5.15.91/arch/x86/entry/syscalls/syscall_64.tbl`

System call names:

0: read

1: write

2: open

3: close

4: stat

5: fstat

6: lstat

39: getpid

120: getresgid

150: munlock

Sample output for *strace ls*:

```
execve("/usr/bin/ls", ["ls"], 0x7ffdb1ed0a30 /* 58 vars */) = 0
brk(NULL)                                = 0x55fbd7256000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffda5113690) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f3052620000
access("/etc/ld.so.preload", R_OK)        = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=61891, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 61891, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f3052610000
close(3)                                  = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libselinux.so.1", O_RDONLY|O_CLOEXEC) =
3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0\0\0"...,
832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=166280, ...}, AT_EMPTY_PATH) =
0
mmap(NULL, 177672, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f30525e4000
mprotect(0x7f30525ea000, 139264, PROT_NONE) = 0
mmap(0x7f30525ea000, 106496, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x6000) = 0x7f30525ea000
mmap(0x7f3052604000, 28672, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x20000) = 0x7f3052604000
mmap(0x7f305260c000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x27000) = 0x7f305260c000
mmap(0x7f305260e000, 5640, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7f305260e000
close(3)                                  = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\237\2\0\0\0\0\0"...,
832) = 832
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,
784, 64) = 784
pread64(3, "\4\0\0\0 \0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0"...,
48, 848) = 48
pread64(3, "\4\0\0\0\24\0\0\0\3\0\0\0GNU\0i8\235HZ\227\223\333\350s\
360\352\223\340."..., 68, 896) = 68
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=2216304, ...}, AT_EMPTY_PATH) =
0
pread64(3, "\6\0\0\0\4\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0@\0\0\0\0\0\0\0"...,
784, 64) = 784
mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f30523bc000
mmap(0x7f30523e4000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x28000) = 0x7f30523e4000
mmap(0x7f3052579000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x1bd000) = 0x7f3052579000
mmap(0x7f30525d1000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x214000) = 0x7f30525d1000
```

```

mmap(0x7f30525d7000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_ANONYMOUS, -1, 0) = 0x7f30525d7000
close(3) = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpcr2-8.so.0", O_RDONLY|O_CLOEXEC) =
3
read(3, "\177ELF\2\1\1\0\0\0\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0\0\0\0\0\0\0"...,
832) = 832
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=613064, ...}, AT_EMPTY_PATH) =
0
mmap(NULL, 615184, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f3052325000
mmap(0x7f3052327000, 438272, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x2000) = 0x7f3052327000
mmap(0x7f3052392000, 163840, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x6d000) = 0x7f3052392000
mmap(0x7f30523ba000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|
MAP_DENYWRITE, 3, 0x94000) = 0x7f30523ba000
close(3) = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) =
0x7f3052322000
arch_prctl(ARCH_SET_FS, 0x7f3052322800) = 0
set_tid_address(0x7f3052322ad0) = 26370
set_robust_list(0x7f3052322ae0, 24) = 0
rseq(0x7f30523231a0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f30525d1000, 16384, PROT_READ) = 0
mprotect(0x7f30523ba000, 4096, PROT_READ) = 0
mprotect(0x7f305260c000, 4096, PROT_READ) = 0
mprotect(0x55fbd689e000, 4096, PROT_READ) = 0
mprotect(0x7f305265a000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY})
= 0
munmap(0x7f3052610000, 61891) = 0
statfs("/sys/fs/selinux", 0x7ffda51136d0) = -1 ENOENT (No such file or
directory)
statfs("/selinux", 0x7ffda51136d0) = -1 ENOENT (No such file or directory)
getrandom("\xcc\x92\x8b\x86\x90\xff\x7a\x7b", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x55fbd7256000
brk(0x55fbd7277000) = 0x55fbd7277000
openat(AT_FDCWD, "/proc/filesystems", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0444, st_size=0, ...}, AT_EMPTY_PATH) = 0
read(3, "nodev\tsysfs\nnodev\ttmpfs\nnodev\tbd"..., 1024) = 393
read(3, "", 1024) = 0
close(3) = 0
access("/etc/selinux/config", F_OK) = -1 ENOENT (No such file or directory)
openat(AT_FDCWD, "/usr/lib/locale/locale-archive", O_RDONLY|O_CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=17392992, ...}, AT_EMPTY_PATH)
= 0
mmap(NULL, 17392992, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f305128b000
close(3) = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=34, ws_col=153, ws_xpixel=0, ws_ypixel=0}) = 0
openat(AT_FDCWD, ".", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) = 3
newfstatat(3, "", {st_mode=S_IFDIR|0775, st_size=4096, ...}, AT_EMPTY_PATH) = 0
getdents64(3, 0x55fbd725dab0 /* 5 entries */, 32768) = 160
getdents64(3, 0x55fbd725dab0 /* 0 entries */, 32768) = 0
close(3) = 0
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x1), ...},
AT_EMPTY_PATH) = 0
write(1, "Makefile syscall_32.tbl syscal"..., 41Makefile syscall_32.tbl
syscall_64.tbl
) = 41
close(1) = 0

```

```
close(2)                = 0
exit_group(0)            = ?
+++ exited with 0 +++
```

time command:

time ls

```
real    0m0,003s
user    0m0,001s
sys     0m0,003s
```

Figure 3: Output of *time ls* command

time cp test.txt test2

```
real    0m0,004s
user    0m0,004s
sys     0m0,000s
```

Figure 4: Output of *time cp test.txt test2* command

time strace ls

```
real    0m0,126s
user    0m0,001s
sys     0m0,015s
```

Figure 5: Output of *time strace ls* command

- 1) **real:** The total time elapsed from the start to the finish of the call.
- 2) **user:** The amount of CPU time spent in user mode (outside the kernel, privileged instructions cannot be executed).
- 3) **sys:** The amount of CPU time spent in the kernel mode within the process (privileged instructions can be executed).

C Program

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/time.h>

struct node {
    int value;
    struct node* next;
    struct node* prev;
};

struct node* createNode(int value) {
    struct node* newNode = (struct node*) malloc(sizeof( struct node));
    newNode->value = value;
    newNode->next = NULL;
    newNode->prev = NULL;
    return newNode;
}

void insert(struct node** head, int value) {
```

```

if ( *head == NULL) {
    *head = createNode( value);
    return;
}
struct node* newNode = createNode(value);
if ( (*head)->value >= value) {
    newNode->next = *head;
    newNode->next->prev = newNode;
    *head = newNode;
}
else {
    struct node* cur = *head;

    while (cur->next && cur->next->value < newNode->value) {
        cur = cur->next;
    }

    newNode->next = cur->next;
    if (cur->next) {
        newNode->next->prev = newNode;
    }
    cur->next = newNode;
    newNode->prev = cur;
}
}

// function to test if the sorted doubly linked list is implemented correctly
void printList(struct node* head) {
    int count = 0;
    while (head) {
        printf("%d\n", head->value);
        head = head->next;
        count++;
    }
    printf("%d", count);
}

int main() {
    struct Node* head = NULL;
    struct timeval start;
    struct timeval end;
    gettimeofday(&start, 0);

    // inserting 10000 random integers into the doubly linked list
    for ( int i = 0; i < 10000; i++) {
        insert(&head, rand());
    }
    gettimeofday(&end, 0);
    printf("It took %f seconds to insert 10000 elements into the list.\n",
(end.tv_usec - start.tv_usec) / 1000000.0);
    printList(head);
    return 0;
}

```


Makefile

```
all: list
list: list.c
    gcc -Wall -g -o list list.c
clean:
    rm -fr list list.o *~
```