CS342 Operating Systems - Spring 2023
Homework #1

2- Since I used various Linux distributions on VMware and VirtualBox before, I had no difficulty with the installation part. Also, when I selected the ubuntu iso, VMware suggested easy-install. This simplifies the installation process. My machine couldn't get the update and ubuntu packages just when I first installed it. This was preventing me from downloading "open-vm-tools" and other tools or updates. So I deleted and reinstalled it.

sudo: Run a command with administrator rights by using sudo.

df: Display disk use statistics with the df command.

cd: Change the current working directory by typing cd.

ls: List the files and directories in the active working directory with the ls command.

ifconfig: Display details about the system's network interfaces with the ifconfig command.

top: Show details about the system's processes and resource utilization.

kill: Put a process to rest.

apt-get: The Ubuntu package manager is called apt-get.

pwd: Print the current working directory by typing pwd.

ps: Display details about the system's processes with the ps command.

3- ubuntu@ubuntu:~$ ls /boot | grep vmlinuz

/boot/vmlinuz

ubuntu@ubuntu:~$ uname -r

5.15.0-43-generic

4-

arch, block, certs, crypto, Documentation, drivers, fs, include, init, ipc, kernel, lib, LICENSES, mm, net, samples, scripts, security, sound, tools, usr, virt

5-

/Users/faruk.ulutas/Downloads/linux-5.4.230/arch/x86/entry/syscalls/syscall_64.tbl

0 read

1 write

2 open

3 close

4 stat

5 fstat

6 lstat

39 getpid

120 getresgid

150 munlock

6-

ubuntu@ubuntu:~/Desktop$ strace ls

execve("/usr/bin/ls", ["ls"], 0x7ffc25be5620 /* 54 vars */) = 0

brk(NULL) = 0x55c1a457d000

arch_prctl(0x3001 /* ARCH_??? */, 0x7ffc8f1e6a10) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8969feb000

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=62475, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 62475, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8969fdb000

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\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) = 0x7f8969faf000

mprotect(0x7f8969fb5000, 139264, PROT_NONE) = 0

mmap(0x7f8969fb5000, 106496, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7f8969fb5000

mmap(0x7f8969fcf000, 28672, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x20000) = 0x7f8969fcf000

mmap(0x7f8969fd7000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x27000) = 0x7f8969fd7000

mmap(0x7f8969fd9000, 5640, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f8969fd9000

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\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\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) = 0x7f8969d87000

mmap(0x7f8969daf000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) = 0x7f8969daf000

mmap(0x7f8969f44000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f8969f44000

mmap(0x7f8969f9c000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) = 0x7f8969f9c000

mmap(0x7f8969fa2000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f8969fa2000

close(3) = 0

openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpcre2-8.so.0", O_RDONLY|O_CLOEXEC) = 3

read(3, "\177ELF\2\1\1\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=613064, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 615184, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7f8969cf0000

mmap(0x7f8969cf2000, 438272, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f8969cf2000

mmap(0x7f8969d5d000, 163840, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6d000) = 0x7f8969d5d000

mmap(0x7f8969d85000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x94000) = 0x7f8969d85000

close(3) = 0

```
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f8969ced000
arch_prctl(ARCH_SET_FS, 0x7f8969ced800) = 0
set_tid_address(0x7f8969cedad0)         = 2768
set_robust_list(0x7f8969cedae0, 24)     = 0
rseq(0x7f8969cee1a0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f8969f9c000, 16384, PROT_READ) = 0
mprotect(0x7f8969d85000, 4096, PROT_READ) = 0
mprotect(0x7f8969fd7000, 4096, PROT_READ) = 0
mprotect(0x55c1a3a09000, 4096, PROT_READ) = 0
mprotect(0x7f896a025000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f8969fdb000, 62475)           = 0
statfs("/sys/fs/selinux", 0x7ffc8f1e6a50) = -1 ENOENT (No such file or directory)
statfs("/selinux", 0x7ffc8f1e6a50)      = -1 ENOENT (No such file or directory)
getrandom("\x34\x70\xcd\x9d\x96\x29\xfb\xa1", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0x55c1a457d000
brk(0x55c1a459e000)                     = 0x55c1a459e000
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) = 378
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=17388592, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 17388592, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f8968c57000
close(3)                                = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=24, ws_col=80, 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|0755, st_size=4096, ...}, AT_EMPTY_PATH) = 0
getdents64(3, 0x55c1a4584ab0 /* 2 entries */, 32768) = 48
getdents64(3, 0x55c1a4584ab0 /* 0 entries */, 32768) = 0
close(3)                        = 0
close(1)                        = 0
close(2)                        = 0
exit_group(0)                       = ?
+++ exited with 0 +++
```

7-

```
ubuntu@ubuntu:~/Desktop$ time ls

real  0m0,002s
user  0m0,002s
sys  0m0,000s
```

Real time, sometimes known as "wall-clock time," is the amount of time required to complete a task.

CPU time spent executing instructions in user mode is referred to as "user" time (i.e., outside the kernel).

The amount of time the CPU spends running kernel mode instructions is known as "sys" time.

8- list.c

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

typedef struct node {
  struct node *next;
```

```
  struct node *prev;
  int data;
} Node;

Node *head = NULL;

Node *create(int data) {
  Node *node = (Node *)malloc(sizeof(Node));

  node->next = NULL;
  node->prev = NULL;
  node->data = data;

  return node;
}

void insert(int data) {
  Node *node = create(data);

  if (head == NULL || head->data >= data) {
    node->next = head;
    head = node;
    return;
  }
Node *current = head;

  while (current->next != NULL && current->next->data < data) {
    current = current->next;
  }

  node->next = current->next;
```

```
  current->next = node;
}

int main() {
  struct timeval start, end;
  int i;
  long elapsed_time;

  gettimeofday(&start, NULL);
  srand(time(NULL));

  for (i = 0; i < 10000; i++) {
    insert(rand());
  }

  gettimeofday(&end, NULL);
  elapsed_time = (end.tv_sec - start.tv_sec) * 1000000 + (end.tv_usec - start.tv_usec);
  printf("It took %ld microseconds to insert 10000 random interegers.\n", elapsed_time);
  return 0;
}
```

Makefile

```
all:
gcc list.c -o list
```