Bilkent University

CS342: Operating Systems

Homework 1

Berdan Akyürek

21600904

Section: 3

1. Installing Ubuntu

In the first part of the homework, I installed Ubuntu 20.04. I installed it on a virtual machine under my current system using VirtualBox. I allocated 25 GB of disk space and 4 GB of memory for this virtual machine.
Installation steps were easy. I simply downloaded the ISO and booted the VM from this ISO. I used the graphical installer and followed the steps. I used the default partitioning instead of creating partitions myself.
Later I started to work on some new commands as stated in the assignment. Since I have been using Linux for a few years, I already know some basic commands. So I tried to learn some more other commands that I did not learn before. Here is a list of the new commands I learned:

- sort: sorts a file in ascending order. -r flag sorts it in descending order.
- awk: a command to process text files
- alias: used to give a short command to a long, hard to type and frequently used command.
- paste: merges two files line by line.
- lpr: prints the specified file(s) using the specified physical printer.
- lsof: list of all open files, the PID's that use that file etc.
- watch: runs a program for each n seconds (2 for default) and shows the outputs.
- pwdx: print working directory for a process with specified PID
- wc: line, word and byte count of file
- uniq: find duplicated lines in a file

2. Kernel Path and Version

Kernel path: /boot/vmlinuz
Kernel version: 5.11.0

3. Kernel Code

Downloaded Kernel version: 5.10.70
Subdirectories of kernel code:

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

4. System Call Table

System call table directory: arch/x86/entry/syscalls/syscall_64.tbl
Requested syscalls:

- 3. close
- 35. nanosleep
- 110. getppid
- 210. io_cancel

5. Strace Command

Sample output of “strace ls” in home directory:

```
execve("/usr/bin/ls", ["ls"], 0x7ffe762322f0 /* 56 vars */) = 0
brk(NULL)                               = 0x55f7d51f3000
arch_prctl(0x3001 /* ARCH_??? */, 0x7ffd1292e890) = -1 EINVAL (Invalid argument)
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
```

```
fstat(3, {st_mode=S_IFREG|0644, st_size=71924, ...}) = 0
mmap(NULL, 71924, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f66caa51000
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@p\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=163200, ...}) = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x7f66caa4f000
mmap(NULL, 174600, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f66caa24000
mprotect(0x7f66caa2a000, 135168, PROT_NONE) = 0
mmap(0x7f66caa2a000, 102400, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x6000) = 0x7f66caa2a000
mmap(0x7f66caa43000, 28672, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1f000) = 0x7f66caa43000
mmap(0x7f66caa4b000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x26000) = 0x7f66caa4b000
mmap(0x7f66caa4d000, 6664, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f66caa4d000
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\0\360q\2\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\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) =
32
pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"
..., 68, 880) = 68
fstat(3, {st_mode=S_IFREG|0755, st_size=2029224, ...}) = 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
pread64(3, "\4\0\0\0\20\0\0\0\5\0\0\0GNU\0\2\0\0\300\4\0\0\0\3\0\0\0\0\0\0\0", 32, 848) =
32
pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\t\233\222%\274\260\320\31\331\326\10\204\276X>\263"
..., 68, 880) = 68
mmap(NULL, 2036952, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f66ca832000
mprotect(0x7f66ca857000, 1847296, PROT_NONE) = 0
mmap(0x7f66ca857000, 1540096, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x25000) = 0x7f66ca857000
```

```
mmap(0x7f66ca9cf000, 303104, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x19d000) = 0x7f66ca9cf000
mmap(0x7f66caa1a000, 24576, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1e7000) = 0x7f66caa1a000
mmap(0x7f66caa20000, 13528, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f66caa20000
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\0\3\0>\0\1\0\0\0\340\"\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=584392, ...}) = 0
mmap(NULL, 586536, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f66ca7a2000
mmap(0x7f66ca7a4000, 409600, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x2000) = 0x7f66ca7a4000
mmap(0x7f66ca808000, 163840, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x66000) = 0x7f66ca808000
mmap(0x7f66ca830000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x8d000) = 0x7f66ca830000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libdl.so.2", 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 \22\0\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0644, st_size=18816, ...}) = 0
mmap(NULL, 20752, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f66ca79c000
mmap(0x7f66ca79d000, 8192, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1000) = 0x7f66ca79d000
mmap(0x7f66ca79f000, 4096, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f66ca79f000
mmap(0x7f66ca7a0000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x3000) = 0x7f66ca7a0000
close(3)                                = 0
openat(AT_FDCWD, "/lib/x86_64-linux-gnu/libpthread.so.0", 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\220\201\0\0\0\0\0\0"..., 832) =
832
pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824)
= 68
fstat(3, {st_mode=S_IFREG|0755, st_size=157224, ...}) = 0
pread64(3,
"\4\0\0\0\24\0\0\0\3\0\0\0GNU\0\345Ga\367\265T\320\374\301V)Yf]\223\337"..., 68, 824)
= 68
```

```
mmap(NULL, 140408, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0x7f66ca779000
mmap(0x7f66ca780000, 69632, PROT_READ|PROT_EXEC,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x7000) = 0x7f66ca780000
mmap(0x7f66ca791000, 20480, PROT_READ,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x18000) = 0x7f66ca791000
mmap(0x7f66ca796000, 8192, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c000) = 0x7f66ca796000
mmap(0x7f66ca798000, 13432, PROT_READ|PROT_WRITE,
MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f66ca798000
close(3)                                = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
-1, 0) = 0x7f66ca777000
arch_prctl(ARCH_SET_FS, 0x7f66ca778400) = 0
mprotect(0x7f66caa1a000, 12288, PROT_READ) = 0
mprotect(0x7f66ca796000, 4096, PROT_READ) = 0
mprotect(0x7f66ca7a0000, 4096, PROT_READ) = 0
mprotect(0x7f66ca830000, 4096, PROT_READ) = 0
mprotect(0x7f66caa4b000, 4096, PROT_READ) = 0
mprotect(0x55f7d45df000, 4096, PROT_READ) = 0
mprotect(0x7f66caa90000, 4096, PROT_READ) = 0
munmap(0x7f66caa51000, 71924)           = 0
set_tid_address(0x7f66ca7786d0)         = 2925
set_robust_list(0x7f66ca7786e0, 24)     = 0
rt_sigaction(SIGRTMIN, {sa_handler=0x7f66ca780bf0, sa_mask=[],
sa_flags=SA_RESTORER|SA_SIGINFO, sa_restorer=0x7f66ca78e3c0}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {sa_handler=0x7f66ca780c90, sa_mask=[],
sa_flags=SA_RESTORER|SA_RESTART|SA_SIGINFO, sa_restorer=0x7f66ca78e3c0},
NULL, 8) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024,
rlim_max=RLIM64_INFINITY}) = 0
statfs("/sys/fs/selinux", 0x7ffd1292e7e0) = -1 ENOENT (No such file or directory)
statfs("/selinux", 0x7ffd1292e7e0)      = -1 ENOENT (No such file or directory)
brk(NULL)                               = 0x55f7d51f3000
brk(0x55f7d5214000)                     = 0x55f7d5214000
openat(AT_FDCWD, "/proc/filesystems", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0444, st_size=0, ...}) = 0
read(3, "nodev\tsysfs\nnodev\ttmpfs\nnodev\tbd"..., 1024) = 360
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
fstat(3, {st_mode=S_IFREG|0644, st_size=17339232, ...}) = 0
```

```
mmap(NULL, 17339232, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f66c96ed000
close(3)                                = 0
ioctl(1, TCGETS, {B38400 opost isig icanon echo ...}) = 0
ioctl(1, TIOCGWINSZ, {ws_row=29, ws_col=79, ws_xpixel=0, ws_ypixel=0}) = 0
openat(AT_FDCWD, ".", O_RDONLY|O_NONBLOCK|O_CLOEXEC|O_DIRECTORY) =
3
fstat(3, {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
getdents64(3, /* 20 entries */, 32768)  = 648
getdents64(3, /* 0 entries */, 32768)   = 0
close(3)                                = 0
fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}) = 0
write(1, "Desktop  Documents  Downloads  M"..., 72Desktop  Documents  Downloads
Music  Pictures    Public    Templates  Videos
) = 72
close(1)                                = 0
close(2)                                = 0
exit_group(0)                             = ?
+++ exited with 0 +++
```

6. Time Command

“time ls” output:
real 0m0.004s
user 0m0.000s
sys 0m0.004s

“time cp deneme.txt temp/deneme.txt” output:
real 0m0.004s
user 0m0.000s
sys 0m0.005s

“time cat deneme.txt” output:
real 0m0.004s
user 0m0.003s
sys 0m0.001s

In this output, real refers to passed real time during execution. User and sys refers to CPU spent on that process. For example, if there are more processes running at the same time, real time should count the time spent for the other processes too. However user and sys should not count this time. In order to test

this, I created a python script which is just an infinite loop. I started this code with running “python3 deneme.py &” 20 times. While all are running, I started “time ls” and here is the output:

real 0m0.017s
user 0m0.001s
sys 0m0.000s

It is possible to see that real time increased significantly and user and sys did not. When there are more processes running, real time increased significantly because some of this 0.017 seconds was spent on other processes except ls. However, user and sys times depend on how much time CPU spent on only ls process. That is why running more processes at the same time does not change these two times significantly.
The difference between user and sys values is, user is time spent on user mode and sys is time spent by kernel syscalls.

7. Linked List Implementation in C and Makefile

C Code:

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

struct Node {
	int value;
	struct Node* next;
};

// adds n at the end of linked list with head
void addToLinkedList(struct Node** head, int n)
{
	struct Node* new = (struct Node*)malloc(sizeof(struct Node));

	new -> next = NULL;
	new -> value = n;

	if(*head == NULL)
	{
```

```
        *head = new;
        return;
        }

        struct Node* current = *head;

        while(current->next != NULL)
        current = current->next;

        current -> next = new;
}

void printLinkedList(struct Node* head)
{
        if(head == NULL)
        {
        printf("Empty linked list.\n");
        return;
        }

        while(head != NULL)
        {
        printf("%d\n", head->value);
        head = head->next;
        }

}

int main()
{
        struct Node* head = NULL;
        srand(time(NULL));

        struct timeval beforeAll;
        gettimeofday(&beforeAll, NULL);

        for(int i = 0; i < 10000; i ++)
        {
        struct timeval beforei;
        gettimeofday(&beforei, NULL);

        addToLinkedList(&head, rand() % 101);

        struct timeval afteri;
```

```
        gettimeofday(&afteri, NULL);

        printf("Execution of adding node %d took %ld seconds and %ldmicroseconds\n",
i
        , afteri.tv_sec - beforei.tv_sec
        , (((afteri.tv_sec - beforei.tv_sec) * 1000000)
        + afteri.tv_usec) - (beforei.tv_usec));
        }

        struct timeval afterAll;
        gettimeofday(&afterAll, NULL);

        //printLinkedList(head);

        printf("All execution took %ld seconds and %ld microseconds\n"
        , afterAll.tv_sec - beforeAll.tv_sec
        , (((afterAll.tv_sec - beforeAll.tv_sec) * 1000000)
        + afterAll.tv_usec) - (beforeAll.tv_usec));

        return 0;
}
```

Makefile:

```
all: list
list: list.c
   gcc -o list list.c
clean:
   rm -f list list.o *~
```