Nama: Daffa Harikhsan NIM: 23/513044/PA/21918

Tugas 2

No. 1 (3.1)

Perintah:

- uname,
- df,
- hostname,
- hostname -1

```
-$ uname
Linux
 _$ df
Filesystem
                               Used Available Use% Mounted on
                1K-blocks
none
                  3912876
                                        3912872
                                                 1% /mnt/wsl
                                      169707612 65% /usr/lib/wsl/drivers
                475781444 306073832
drivers
                  3912876
                                        3912876
                                                  0% /usr/lib/modules
none
                                  0
none
                  3912876
                                        3912876
                                                  0% /usr/lib/modules/5.15.153.1-microsoft-standard-WSL2
               1055762868
                                                  1% /
/dev/sdc
                              747896 1001311500
none
                  3912876
                                        3912800
                                                  1% /mnt/wslg
                                                  0% /usr/lib/wsl/lib
1% /init
none
                  3912876
                                        3912876
rootfs
                  3909524
                                2080
                                        3907444
                                                  0% /dev
                  3909524
                                        3909524
none
                                  0
                                                  0% /run
0% /run/lock
none
                  3912876
                                        3912876
                  3912876
                                  0
                                        3912876
none
none
                  3912876
                                        3912876
                                                  0% /run/shm
                                                  0% /run/user
                  3912876
                                        3912876
none
                  3912876
                                  0
                                        3912876
                                                  0% /sys/fs/cgroup
tmpfs
                  3912876
                                        3912784
                                                  1% /mnt/wslg/versions.txt
none
                  3912876
                                 92
                                        3912784
                                                  1% /mnt/wslg/doc
none
C:\
                475781444 306073832 169707612 65% /mnt/c
 -$ hostname
Daffa-Laptop
   ·(pegoo⊛ Daffa-Laptop) - [~]
 -$ hostname-i
-bash: hostname-i: command not found
```

No. 2 (3.2)

1. Save the trace of a command echo hello to a file titled "echo.log". dharikhsan@cloudshell:~\$ strace -o echo.log echo hello hello

```
dharikhsan@cloudshell:~$ cat echo.log
execve("/usr/bin/echo", ["echo", "hello"], 0x7ffd6e6f08a8 /* 61 vars */) = 0
 execve("/usr/bin/ecno", ["ecno", "nello"], Ux/lindeerOsas /* 61 vars */) = 0
brk(NULL) = 0x55a891bb3000

arch prctl(0x3001 /* ARCH ??? */, 0x7ffcebd77fa0) = -1 EINVAL (Invalid argument)

mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x799671c95000

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=36503, ...}, AT_EMPTY_PATH) = 0

mmap(NULL, 36503, PROT_READ, MAP_PRIVATE, 3, 0) = 0x799671c8c000

close(3) = 0
 crose(s)
mmap (NULL, 12288, PROT READ|PROT WRITE, MAP
arch_prctl(ARCH_SET_FS, 0x799671a60740) = 0
set_tid_address(0x799671a60410) = 0
                                                   PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x799671a60000
mmap(map()
arch prctl (ARCH_SET_FS, 0A)
set_tid_address(0x799671a60a10) = 876
set_tid_address(0x799671a60a20, 24) = 0
rseq(0x799671a610e0, 0x20, 0, 0x53053053) = 0
mprotect(0x799671c79000, 16384, PROT_READ) = 0
mprotect(0x55a891724000, 4096, PROT_READ) = 0
mprotect(0x799671c7000, 8192, PROT_READ) = 0
prlimit64(10, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x799671c8c000, 36503) = 0
munmap(0x799671c8c000, 36503) = 0
setrandom("\xc0\xsc\x7e\xbf\xf6\x65\x23\x16", 8, GRND_NONBLOCK) = 8
brk(NULL) = 0x55a891bd3000
sox5a891bd4000

- 0x55a891bd4000
- 0x55a891bd4000
- 0x55a891bd4000
- 0x55a891bd4000
- 0x55a891bd4000
- 0x55a891bd4000
  openat(AT FDCWD, "/usr/lib/locale/locale-archive", O RDONLY|O CLOEXEC) = 3
newfstatat(3, "", {st_mode=S_IFREG|0644, st_size=3048928, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 3048928, PROT_READ, MAP_PRIVATE, 3, 0) = 0x799671777000
  newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0x2), ...}, AT_EMPTY_PATH) = 0
write(1, "hello\n", 6) = 6
close(1) = 0
  close(2)
    +++ exited with 0 +++
```

```
2. Filter the "echo.log" file to find the text "hello" being mentioned. Screen capture the result. dharikhsan@cloudshell:~$ grep hello echo.log
 execve("/usr/bin/echo", ["echo", "hello"], 0x7ffd6e6f08a8 /* 61 vars */) = 0
 write(1, "hello\n", 6)
 dharikhsan@cloudshell:~$
```

3. Explain what the system call that related to the text in question number 2 is doing based on the manual page of the system call.

Answer:

Echo adalah sebuah perintah untuk memprint sebuah perintah ke layar contohnya jika di python itu print, dan di C++ yaitu cout. Perintah "echo" sendiri di dalam perintah "grep hello echo.log" dijelaskan bagaimana isi dari file echo.log itu muncul di layar. Dalam konteks "grep hello echo.log" ada sebuah panggilan sistem yang disebut "write". "Write" adalah sebuah fungsi inti sistem operasi yang berfungsi untuk menulis data ke suatu file descriptor. File descriptor dapat berupa file, perangkat, atau aliran data seperti output ke layar (print).

Write bekerja dalam perintah "grep hello echo.log" yaitu:

Perintah grep akan mencari kata "hello" di dalam file "echo.log". jika grep mendapatkan kata hello tersebut, grep menyiapkan data yang akan ditampilkan.

Write:

Grep kemudian memanggil write dengan perintah atau argumen file descriptor (fd) (merepresentasikan output standar, yakni layar). Setelat itu ada buffer (pointer ke lokasi memori yang berisi data yang akan ditulis). Dan ada jumlah byte (jumlah karakter yang akan ditulis, dalam kasus ini adalah panjang dari string).

Ditampilkan di layar: Sistem operasi akan mengambil data dari buffer yang diberikan Baris perintah tersebut menjelaskan bahwa:

- Execve = sebuah panggilan sistem untuk menjalankan program baru
- /usr/bin/echo = directory file yang akan dijalankan, yakni perintah echo
- ["echo", "hello"] = argumen yang akan diteruskan ke perintah echo. Argumen pertama = perintah itu sendiri, argumen kedua = string "hello" yang akan dicetak
- 0x7ffd6e6f08a8 = alamat memori
- /* 61 vars */ = komentar yang menunjukkan ada 61 variabel dalam array
- = 0 = nilai pengembalian dari panggilan execve, dan nilai 0 menunjukkan bahwa panggilan berhasil dan program baru sedang berjalan

write(1, "hello\n", 6)

Baris perintah tersebut menjelaskan bahwa:

- Write = nama panggilan sistem, dan digunakan untuk menulis data ke file deskriptor
- $(1, \text{``hello} \ n'', 6) =$
 - 1 = merupakan file deskriptor, 1 merujuk ke stdout (print) yang biasanya adalah layar.
 - "hello\n" = ini adalah string yang akan ditulis, "\n" akan membuat newline jadi seperti enter dalam keyboard
 - 6 = ini adalah jumlah byte yang akan ditulis sesuai dengan "hello\n"