CS347 Lab-1

1.

a.

Using lscpu command:

Architecture: x86_64
Address sizes: 39 bits physical, 48 bits virtual
Byte Order: Little Endian

Similar information is present also in file /proc/cpuinfo, and “grep -o -w 'lm' /proc/cpuinfo” (for architecture).

b.

Using lscpu:

CPU sockets: 1
CPU cores per socket: 6
CPUs: 12

c.

Using lscpu:

Caches (sum of all):
L1d: 192 KiB (6 instances)
L1i: 192 KiB (6 instances)
L2: 1.5 MiB (6 instances)
L3: 12 MiB (1 instance)

d.

cat /proc/meminfo

MemTotal: 8057640 kB
MemFree: 2548432 kB
MemAvailable: 5237760 kB

We can also use the ‘free’ command.

Secondary Memory:

df -Th

Filesystem	Type	Size	Used	Avail	Use%	Mounted on
/dev/sda5	ext4	28G	11G	16G	42%	/ (root partition)
/dev/nvme0n1p1	vfat	256M	36M	221M	14%	/boot/efi
/dev/sda7	ext4	28G	2.4G	24G	9%	/home
/dev/sda4	fuseblk	253G	13G	241G	6%	/media/kabra/GD

/dev/sda3 fuseblk 400G 114G 287G 29% /media/kabra/SK, VK.

e.

man ps gives commands to see all processes: ps -e, or ps aux.

ps aux

a = show processes for all users
u = display the process's user/owner
x = also show processes not attached to a terminal

Total number of processes:
ps aux | wc -l
339

To see state: under header “STAT”
ps -e -o stat | grep <required state code> | wc -l

Running processes:
STAT = R
Count = 2

Sleeping:
STAT = S

```
kabra@IdeaPad-L340:~$ ps -e -o stat | grep S | wc -l
243
```

Stopped:
STAT = T or t

```
kabra@IdeaPad-L340:~$ ps -e -o stat | grep -E '[Tt]' | wc -l
1
```

Used regex with grep (-E)

Zombie:

```
kabra@IdeaPad-L340:~$ ps -e -o stat | grep Z | wc -l
0
```

f.
Number of context switches performed by the system since bootup: pid=1 is the init process primarily responsible for starting and shutting down the system.

```
kabra@IdeaPad-L340:/media/kabra/GD/GD/5/CS333_OS/CS347_Lab/lab1/lab1/myos$ cat /proc/1/sched | grep switches
nr_switches                                  :                 9047
nr_voluntary_switches                        :                 8519
nr_involuntary_switches                      :                  528
kabra@IdeaPad-L340:/media/kabra/GD/GD/5/CS333_OS/CS347_Lab/lab1/lab1/myos$ grep ctxt /proc/1/status
voluntary_ctxt_switches:        8519
nonvoluntary_ctxt_switches:     528
```

Commands:

cat /proc/1/sched | grep switches, or
grep ctxt /proc/1/status

Both commands give the same result. For the former, fields are ‘nr_voluntary_switches’, ‘nr_involuntary_switches’.

2.

To get VmSize, VmRSS:
command used: cat /proc/<pid>/status | grep Vm

memory_1.c
VmSize: 6560 kB
VmRSS: 4740 kB
memory_2.c
VmSize: 10464 kB
VmRSS: 8776 kB
memory_3.c
VmSize: 6560 kB
VmRSS: 4680 kB
memory_4.c
VmSize: 6556 kB
VmRSS: 4840 kB

1 and 2:
ARRAY_SIZE being 1000000 and 2000000 creates the difference.

1, 3 and 4:
VmSize is the same, as ARRAY_SIZE is same for all 3. But VmRSS (cumulative memory) differs, as there is extra code (for-loop) in 3 and 4.

Other related commands:

command time -v ./a.out
memusage -T ./a.out

3.

```
kabra@IdeaPad-L340:/media/kabra/GD/GD/5/CS333_OS/CS347_Lab/lab1/lab1/subprocess$ ps aux | grep "./subprocesses 200050157"
kabra      30850  0.0  0.0   2764   936 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30851  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30852  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30853  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30854  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30855  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30856  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30857  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30858  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30859  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30860  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30861  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30862  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30863  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30864  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30865  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30866  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30867  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30868  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30869  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30870  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30871  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30872  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30873  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30874  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30875  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30876  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30877  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30878  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30879  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30880  0.0  0.0   2764    96 pts/1    S+   07:26   0:00 ./subprocesses 200050157
kabra      30894  0.0  0.0  17872  2308 pts/2    S+   07:26   0:00 grep --color=auto ./subprocesses 200050157
```

31 processes.
Command used:

ps aux | grep "./subprocesses 200050157" | wc -l

Output = 32

This gives ps output containing the string “./subprocesses 200050157”. This also includes a process “grep –color=auto”, so 1 has been subtracted from the total number.

4.

“2>file” redirects stderr output to file

Commands used:
strace ./empty 2>empty_error.txt
strace ./hello 2>empty_hello.txt

Output for empty:

```
execve("./empty", ["./empty"], 0x7fff0441fe00 /* 48 vars */) = 0
brk(NULL)                               = 0x2273000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fffe17ca6a0) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f0ac1811000
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=66853, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 66853, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7f0ac1800000
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\0\211\303\313\205\371\345PFwdq\376\320^\304A"..., 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) = 0x7f0ac15d8000
```

```
mmap(0x7f0ac1600000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7f0ac1600000
mmap(0x7f0ac1795000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7f0ac1795000
mmap(0x7f0ac17ed000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) =
0x7f0ac17ed000
mmap(0x7f0ac17f3000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7f0ac17f3000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7f0ac15d5000
arch_prctl(ARCH_SET_FS, 0x7f0ac15d5740) = 0
set_tid_address(0x7f0ac15d5a10)         = 16463
set_robust_list(0x7f0ac15d5a20, 24)     = 0
rseq(0x7f0ac15d60e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7f0ac17ed000, 16384, PROT_READ) = 0
mprotect(0x600000, 4096, PROT_READ)     = 0
mprotect(0x7f0ac184b000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7f0ac1800000, 66853)           = 0
exit_group(0)                           = ?
+++ exited with 0 +++
```

Output for hello:

```
execve("./hello", ["./hello"], 0x7ffdb1e67140 /* 48 vars */) = 0
brk(NULL)                               = 0xc75000
arch_prctl(0x3001 /* ARCH_??? */, 0x7fff9b728020) = -1 EINVAL (Invalid argument)
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7feb7b54d000
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=66853, ...}, AT_EMPTY_PATH) = 0
mmap(NULL, 66853, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7feb7b53c000
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"..., 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"..., 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\0\211\303\313\205\371\345PFwdq\376\320^\304A"..., 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"..., 784, 64) = 784
mmap(NULL, 2260560, PROT_READ, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7feb7b314000
mmap(0x7feb7b33c000, 1658880, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x28000) =
0x7feb7b33c000
mmap(0x7feb7b4d1000, 360448, PROT_READ, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1bd000) = 0x7feb7b4d1000
mmap(0x7feb7b529000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x214000) =
0x7feb7b529000
mmap(0x7feb7b52f000, 52816, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7feb7b52f000
close(3)                                = 0
mmap(NULL, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7feb7b311000
arch_prctl(ARCH_SET_FS, 0x7feb7b311740) = 0
set_tid_address(0x7feb7b311a10)         = 16495
set_robust_list(0x7feb7b311a20, 24)     = 0
rseq(0x7feb7b3120e0, 0x20, 0, 0x53053053) = 0
mprotect(0x7feb7b529000, 16384, PROT_READ) = 0
mprotect(0x600000, 4096, PROT_READ)     = 0
mprotect(0x7feb7b587000, 8192, PROT_READ) = 0
prlimit64(0, RLIMIT_STACK, NULL, {rlim_cur=8192*1024, rlim_max=RLIM64_INFINITY}) = 0
munmap(0x7feb7b53c000, 66853)           = 0
getpid()                                = 16495
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
getrandom("\x3e\x20\xc7\x55\xd0\xa0\x77\x9c", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0xc75000
brk(0xc96000)                           = 0xc96000
write(1, "\n", 1)                       = 1
write(1, "Process ID : 16495 \n", 20)   = 20
write(1, "\n", 1)                       = 1
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
write(1, "Enter your name : ", 18)      = 18
read(0, "Virendra\n", 1024)             = 9
write(1, "\n", 1)                       = 1
write(1, "Welcome Virendra\n", 17)      = 17
lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Illegal seek)
exit_group(0)                           = ?
+++ exited with 0 +++
```

Starting part of both outputs is the same.

The difference in outputs is:

```
getpid()                                = 16495
```

```
newfstatat(1, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
getrandom("\x3e\x20\xc7\x55\xd0\xa0\x77\x9c", 8, GRND_NONBLOCK) = 8
brk(NULL)                               = 0xc75000
brk(0xc96000)                           = 0xc96000
write(1, "\n", 1)                       = 1
write(1, "Process ID : 16495 \n", 20)   = 20
write(1, "\n", 1)                       = 1
newfstatat(0, "", {st_mode=S_IFCHR|0620, st_rdev=makedev(0x88, 0), ...}, AT_EMPTY_PATH) = 0
write(1, "Enter your name : ", 18)      = 18
read(0, "Virendra\n", 1024)             = 9
write(1, "\n", 1)                       = 1
write(1, "Welcome Virendra\n", 17)      = 17
lseek(0, -1, SEEK_CUR)                  = -1 ESPIPE (Illegal seek)
```

Some of the common calls correspond to importing libraries (e.g., access("/etc/ld.so.preload", R_OK)) and initializing the memory to run the program (mmap).

Some of the extra calls in ‘hello’ are getpid(), getrandom(), write(), read(), etc. Some of these are the system calls made for the interaction (Enter your name, and then Welcome <name>) [and hence absent from empty’s output].

5.

To get the pid:

```
./openfiles
ps aux | grep openfiles          (it is ps aux | grep {process-name})
```

To get the opened files (open file descriptors):
Command used:

```
ls -l /proc/16745/fd
```

Output:

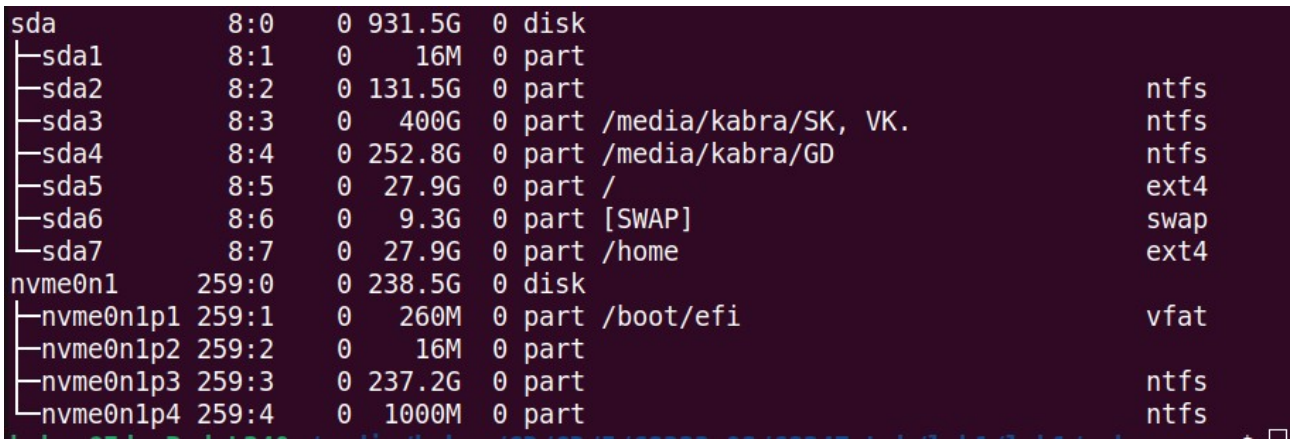
```
total 0
lrwx------ 1 kabra kabra 64 Aug  1 16:32 0 -> /dev/pts/0
lrwx------ 1 kabra kabra 64 Aug  1 16:32 1 -> /dev/pts/0
lrwx------ 1 kabra kabra 64 Aug  1 16:32 2 -> /dev/pts/0
l-wx------ 1 kabra kabra 64 Aug  1 16:32 3 -> '/tmp/welocme to OS'
l-wx------ 1 kabra kabra 64 Aug  1 16:32 4 -> /tmp/CS333
l-wx------ 1 kabra kabra 64 Aug  1 16:32 5 -> /tmp/CS347
```

‘lsof’ can also be used.

6.

lsblk -o +FSTYPE

```
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS                        FSTYPE
```

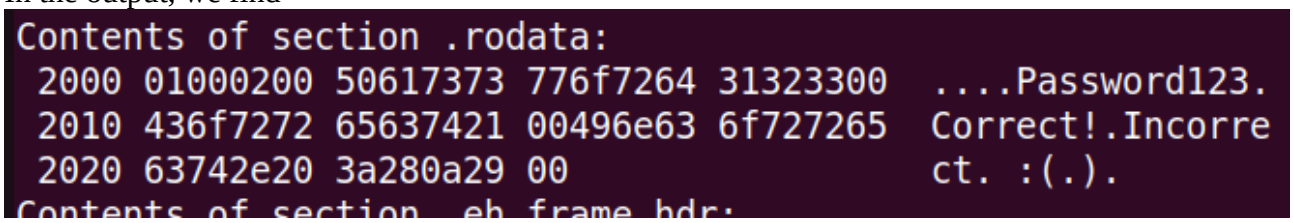


Similar information is also obtained via cat /proc/partitions

7.

objdump -s password.out

In the output, we find



The password is “Password123”.

Answer for Part-2:

1.
- The difference is due to “dw 0xaa55” being present in boot_sector2.asm, but not in boot_sector1.asm.
- The boot disk is detected via this magic number stored as the last two bytes of the boot sector of a disk.
- Since this is absent in the 1st file, we get “No bootable device”. And it is present in the 2nd file (we explicitly wrote the assembly code for that), so the boot disk is recognized, and we get “Booting from Hard Disk...”.

2.
Code: hello.asm
Screenshot: hello.png