DAT470/DIT065 Assignment 1

Michael Hoon
yongm@chalmers.se

Palinya Sengdalavong
palinya_sengdalavong@mymail.sutd.edu.sg

April 24, 2025

Problem 1

(a) Log in to `minerva`

Command:

```
ssh yongm@minerva.cse.chalmers.se
```

Result:

```
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-55-generic x86_64)
* Documentation:  https://help.ubuntu.com
* Management:  https://landscape.canonical.com
* Support:  https://ubuntu.com/pro
System information as of Wed Mar 26 10:44:06 AM CET 2025
System load:  0.11 Swap usage:  0% Users logged in:  6
Usage of /:  1.9% of 1.71TB Temperature:  70.0 C
Memory usage:  59% Processes:  674
* Strictly confined Kubernetes makes edge and IoT secure.  Learn how
MicroK8s just raised the bar for easy, resilient and secure K8s cluster
deployment.  https://ubuntu.com/engage/secure-kubernetes-at-the-edge
Expanded Security Maintenance for Applications is not enabled.
140 updates can be applied immediately.
To see these additional updates run:  apt list --upgradable
5 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm
Last login:  Wed Mar 26 10:42:02 2025 from 129.16.31.115
```

(b) Install Linux Kernel Version 6.13.5

Command:

```
wget https://cdn.kernel.org/pub/linux/kernel/v6.x/linux-6.13.5.tar.xz
```

Result: Installed to `linux-6.13.5.tar.xz` file in `$HOME`.

(c) Extract Tarball Contents

Command: `tar -xvf linux-6.13.5.tar.xz`

Result: Extracted to `linux-6.13.5` Folder in `$HOME`.

(d) Tarball and File Content Size

To determine the tarball size in human-readable format, we use:

Command: `ls -lh`

Result:
```
total 142M
drwxrwxr-x 26 yongm yongm 42 Feb 27 13:34 linux-6.13.5
-rw-rw-r-- 1 yongm yongm 142M Feb 27 13:49 linux-6.13.5.tar.xz
```

From here, we can see that the tarball size before extraction is `142M` (Megabytes) and the size of the directory entry `linux-6.13.5` is `42` (Bytes). To determine the actual size of the contents in the file, we use:

Command: `du -sh linux-6.13.5`

Result: `574M linux-6.13.5`

Here we see that the actual size of the extracted folder is `574M` (Megabytes).

(e) Determine total number of header and source files

Header Files

Command: `find linux-6.13.5 -type f -name "*.h" | wc -l`

Result: `25320`

Source Files

Command: `find linux-6.13.5 -type f -name "*.c" | wc -l`

Result: `34787`

(f) Find the source file that is the longest

First, we change directory to `linux-6.13.5`, then we execute the following command:

Command:

```
find .  -type f -name '*.c' -print0 | xargs -0 wc -l | grep -v 'total'
| sort -nr | head -n 1
```

Result:
`57159 linux-6.13.5/drivers/net/wireless/realtek/rtw89/rtw8852c_table.c`

Here we see that the source file `"rtw8852c_table.c"` has the most number of lines (57159).

(g) Names of 10 Most Included Headers in Kernel Sources

Command:

```
find linux-6.13.5 -type f \( -name "*.c" -o -name "*.h" \) -print0 | \
xargs -0 grep -hE '^#include [<"]([^">]+)[">]' | \
awk '{print $2}' | tr -d '<>"' | sort | uniq -c | sort -nr | head -n 10
```

Result:

```
14087 linux/module.h
9143 linux/kernel.h
9007 linux/slab.h
6942 linux/types.h
6275 linux/init.h
6078 linux/platform_device.h
5392 linux/delay.h
4480 linux/of.h
4315 linux/interrupt.h
4262 linux/io.h
```

(h) Download & Copy `meps.csv` File

After downloading the file into our local device, on a separate terminal tab on the local device, we run the following command:

Command:
`scp $HOME/Download/meps.csv yongm@minerva.cse.chalmers.se:/data/users/yongm`

Result: `meps.csv 100% 915KB 9.5MB/s 00:00`

(i) Number of MEPs Sweden had in the 10th European parliament.

We need to determine the number of MEPs Sweden had in the 10th European parliament, i.e. where the "electoral district" column is "Sweden" and the "parliamentary term" column is "10". We will use the `awk` command for this:

Command:

```
awk -F';' 'NR > 1 && $6 == "Sweden" && $7 == 10 { count++ } END { print
count }' meps.csv
```

Result: `21`

(j) Name of the largest parliamentary group in the 10th European Parliament

Command:

```
awk -F';' 'NR > 1 && $7 == 10 {print $8}' meps.csv | sort | uniq -c
| sort -nr
```

Result:

```
188 European People's Party group
136 Progressive Alliance of Socialists and Democrats
86 Patriots for Europe
79 European Conservatives and Reformists
77 Renew Europe
53 The Greens-European Free Alliance
45 European United Left-Nordic Green Left
32 Non-Inscrits
25 Europe of Sovereign Nations
```

Here we see that **European People's Party Group** has 188 members, making it the largest.

(k) Names and Birth dates of the oldest and youngest MEPs

This can be done by sorting the dataset based on the MEPs' date of birth (DoB) in ascending order. Since the DoB is incremental, the first row will be the oldest MEP while the last row will be the youngest MEP.

Command:

```
awk -F';' 'NR>1 && $5 != "" {print $5 "," $1}' meps.csv | sort | head -n 1

awk -F';' 'NR>1 && $5 != "" {print $5 "," $1}' meps.csv | sort | tail -n 1
```

Result:

```
1893-01-25,Louise Weiss
2001-01-08,Lena Schilling
```

From above, we see that the oldest MEP to serve in the European Parliament is **Louise Weiss**, born 25th January 1893, while the youngest is **Lena Schilling**, born 8th January 2001.

(l) Fraction of Female MEPs out of all MEPs who ever Served

Command:

```
awk -F';' 'NR>1 {count[$4]++} END {print "Fraction of female MEPs:",
count["female"] / (count["female"] + count["male"])}' meps.csv
```

Result:

```
Fraction of female MEPs:  0.296875
```

Problem 2: Information of Computers

We will be choosing the `io` and `uranus` computer in the `minerva` cluster. We will need the following information about the system:

- The model of and the clock frequency of the CPU
- The number of physical CPUs (sockets in use), the number of cores, and the number of hardware threads
- The instruction set architecture of the CPU
- The cache line length
- The amount of L1, L2, and L3 cache
- The amount of system RAM
- The number of GPUs and model of the GPU(s)
- The amount of RAM on the GPU(s)
- The type of filesystem of `/data`
- The total amount of disk space and the amount of free space on `/data`
- The version of the Linux kernel running on the system and the GNU/Linux distribution and its version running on the system
- The filename and the version of the default Python 3 interpreter available on the system (globally installed)

We will first go through the individual commands for each of the information, and construct the compiled shell script after. For each of the information, we group them into the following: CPU, GPU, Memory, Storage, OS, and Python, creating a function that returns all the information for each group.

CPU Information

We know that the command `lscpu` prints information about the CPUs of a system, as well as other important hardware information such as thread and core counts. Here, we use the command to obtain the first 5 listed information:

```
lscpu | grep -E 'Model name|Socket|Core|Thread|MHz|Architecture|L[123]
cache|L1d cache|L1i cache|Level'
```

This pattern filters out all the information we need. However for the cache line length, this is not displayed in `lscpu`, and we need a separate command which "displays the value of a specified configuration variable on the standard output":

```
getconf LEVEL1_DCACHE_LINESIZE
```

GPU Information

`minerva` does not have a GPU, and we need a single script to output all the information we need. Thus, we will create a conditional check if the `nvidia-smi` command runs on the system, then print out the appropriate values. Otherwise, return "No NVIDIA GPU Detected".

```
if command -v nvidia-smi &>/dev/null; then
   nvidia-smi --query-gpu=name,memory.total --format=csv
else
    echo "No NVIDIA GPU Detected"
fi
```

Memory Information

The `free` command allows us to monitor system memory usage:

```
free -h | grep Mem
```

Storage Information

The `df` command displays information about total space and available space on a file system, and we show the type of filesystem in human-readable format:

```
df -hT /data | awk 'NR==1 || NR==2'
```

OS Information

`uname` and `lsb_release` gives us basic system information, which we use to find OS information:

```
uname -srmo
lsb_release -d
```

Python Information

```
if command -v python3 &>/dev/null; then
  python3 --version
  readlink -f $(which python3)
else
  echo "Python3 not found"
fi
```

Output

The final script can be found in the file `assignment1_problem2.sh`. Since the task requires us to run the script on `minerva` and the two computers, we use the following commands (a total of 3 times):

minerva

```
chmod +x assignment1_problem2.sh    \\ set execute permission on the file

./assignment1_problem2.sh
```

io

```
sbatch -w io --output=io_info.txt assignment1_problem2.sh    \\ specify io node
```

uranus

```
sbatch -w uranus --output=io_info.txt assignment1_problem2.sh    \\ specify uranus node
```

minerva Information

```
==== System Information for minerva ====

==== CPU Information ====
Architecture:                         x86_64
Model name:                           Intel(R) Xeon(R) Silver 4410Y
Thread(s) per core:                   2
Core(s) per socket:                   12
Socket(s):                            1
CPU(s) scaling MHz:                   24%
CPU max MHz:                          3900.0000
CPU min MHz:                          800.0000
L1d cache:                            576 KiB (12 instances)
L1i cache:                            384 KiB (12 instances)
L2 cache:                             24 MiB (12 instances)
L3 cache:                             30 MiB (1 instance)
Cache Line Length: 64 bytes

==== Memory Information ====

Mem:           125Gi        72Gi        48Gi       5.8Mi       6.9Gi        52Gi

==== GPU Information ====

No NVIDIA GPU Detected

==== Storage Information ====

Filesystem     Type  Size  Used Avail Use% Mounted on
data           zfs    59T  946G   58T   2% /data

==== OS Information ====

Linux 6.8.0-55-generic x86_64 GNU/Linux
Description:    Ubuntu 24.04.1 LTS

==== Python Information ====

Python 3.12.9
/data/users/yongm/miniforge3/bin/python3.12
```

io Information

```
=== System Information for io ====

==== CPU Information ====

Architecture:                         x86_64
Model name:                           INTEL(R) XEON(R) GOLD 6548N
Thread(s) per core:                   2
Core(s) per socket:                   32
Socket(s):                            2
L1d cache:                            3 MiB (64 instances)
L1i cache:                            2 MiB (64 instances)
L2 cache:                             128 MiB (64 instances)
L3 cache:                             120 MiB (2 instances)
Cache Line Length: 64 bytes

==== Memory Information ====

Mem:           503Gi       7.3Gi       399Gi       6.1Mi       100Gi       496Gi

==== GPU Information ====

name, memory.total [MiB]
NVIDIA L4, 23034 MiB
NVIDIA L4, 23034 MiB
NVIDIA L4, 23034 MiB
NVIDIA L4, 23034 MiB
NVIDIA L4, 23034 MiB
NVIDIA L4, 23034 MiB
NVIDIA L4, 23034 MiB
NVIDIA L4, 23034 MiB

==== Storage Information ====

Filesystem     Type  Size  Used Avail Use% Mounted on
minerva:/data  nfs4   59T  946G   58T   2% /data

==== OS Information ====

Linux 6.8.0-55-generic x86_64 GNU/Linux
Description:    Ubuntu 24.04.2 LTS

==== Python Information ====

Python 3.12.9
/data/users/yongm/miniforge3/bin/python3.12
```

uranus Information

```
==== System Information for uranus ====

==== CPU Information ====

Architecture:                         x86_64
Model name:                           INTEL(R) XEON(R) GOLD 6548N
Thread(s) per core:                   2
Core(s) per socket:                   32
Socket(s):                            2
L1d cache:                            3 MiB (64 instances)
L1i cache:                            2 MiB (64 instances)
L2 cache:                             128 MiB (64 instances)
L3 cache:                             120 MiB (2 instances)
Cache Line Length: 64 bytes

==== Memory Information ====

Mem:           503Gi       6.5Gi       450Gi        11Mi        49Gi       496Gi

==== GPU Information ====

name, memory.total [MiB]
NVIDIA L40S, 46068 MiB
NVIDIA L40S, 46068 MiB
NVIDIA L40S, 46068 MiB
NVIDIA L40S, 46068 MiB

==== Storage Information ====

Filesystem     Type  Size  Used Avail Use% Mounted on
minerva:/data  nfs4   59T  946G   58T   2% /data

==== OS Information ====

Linux 6.8.0-55-generic x86_64 GNU/Linux
Description:    Ubuntu 24.04.2 LTS

==== Python Information ====

Python 3.12.9
/data/users/yongm/miniforge3/bin/python3.12
```

Compiled System Information of Minerva, IO and Uranus

System Info	Minerva	IO	Uranus
CPU Model	Intel(R) Xeon(R) Silver 4410Y	Intel(R) Xeon(R) GOLD 6548N	Intel(R) Xeon(R) GOLD 6548N
Frequency (MHz)	3900.0000	N.A	N.A
No. of CPUs	1	2	2
No. of Cores	12	32	32
Threads per core	2	2	2
Architecture	x86_64	x86_64	x86_64
Cache Length (bytes)	64	64	64
L1d Cache	576 KiB	3 MiB	3 MiB
L1i Cache	384 KiB	2 MiB	2 MiB
L2 Cache	24 MiB	128 MiB	128 MiB
L3 Cache	30 MiB	120 MiB	120 MiB
Sys RAM	125 GiB	503 GiB	503 GiB
GPU No.	N.A	8	4
GPU Model	N.A	NVIDIA L4	NVIDIA L40S
GPU RAM	N.A	23034 MiB	46048 MiB
Filesystem Type	zfs	nfs4	nfs4
Disk Space	59T	59T	59T
Used Space	946G	946G	946G
Free Space	58T	58T	58T
Linux Ver.	6.8.0-55-generic	6.8.0-55-generic	6.8.0-55-generic
Linux Distribution	Ubuntu	Ubuntu	Ubuntu
Linux Distribution Ver.	24.04.1 LTS	24.04.1 LTS	24.04.1 LTS
Python 3 Interpreter	python3.12	python3.12	python3.12
Interpreter Ver.	3.12.9	3.12.9	3.12.9

We note that for `io` and `uranus`, the `lscpu` command does not display the frequency value, but only the `BogoMIPS` value for the CPU.