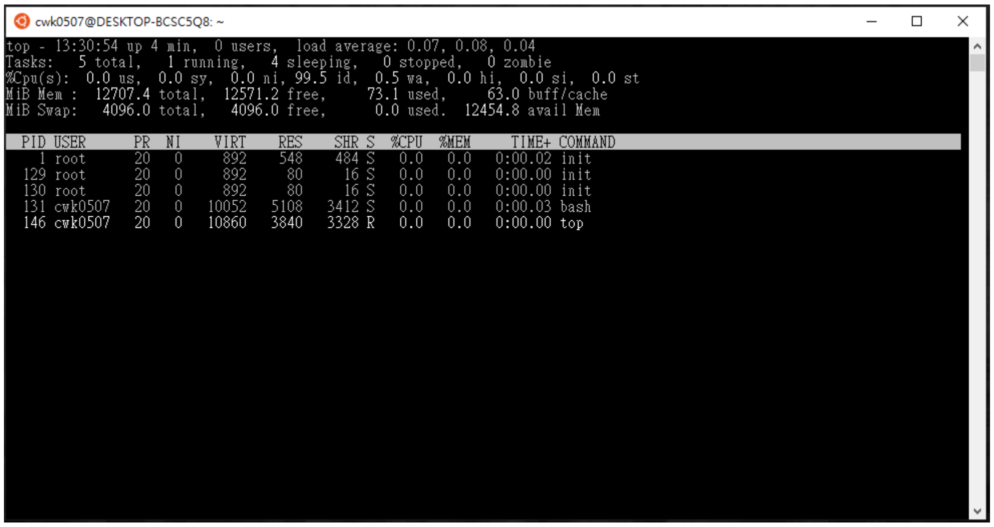
Version of my operating system: Windows 10 Home

Version of bash: 5.0.17(1)

Version of Python: 3.8.2

1. Linux operating system and memory hierarchy

1. Open a terminal, run the command "top", and save a screenshot in your report.



cwk0507@DESKTOP-BCSC5Q8: ~

```
top - 13:30:54 up 4 min,  0 users,  load average: 0.07, 0.08, 0.04
Tasks:   5 total,   1 running,   4 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.0 us,  0.0 sy,  0.0 ni, 99.5 id,  0.5 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  12707.4 total,  12571.2 free,     73.1 used,     63.0 buff/cache
MiB Swap:   4096.0 total,   4096.0 free,      0.0 used.  12454.8 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    1 root      20   0     892    548    484 S   0.0   0.0   0:00.02 init
  129 root      20   0     892     80     16 S   0.0   0.0   0:00.00 init
  130 root      20   0     892     80     16 S   0.0   0.0   0:00.00 init
  131 cwk0507   20   0   10052   5108   3412 S   0.0   0.0   0:00.03 bash
  146 cwk0507   20   0   10860   3840   3328 R   0.0   0.0   0:00.00 top
```

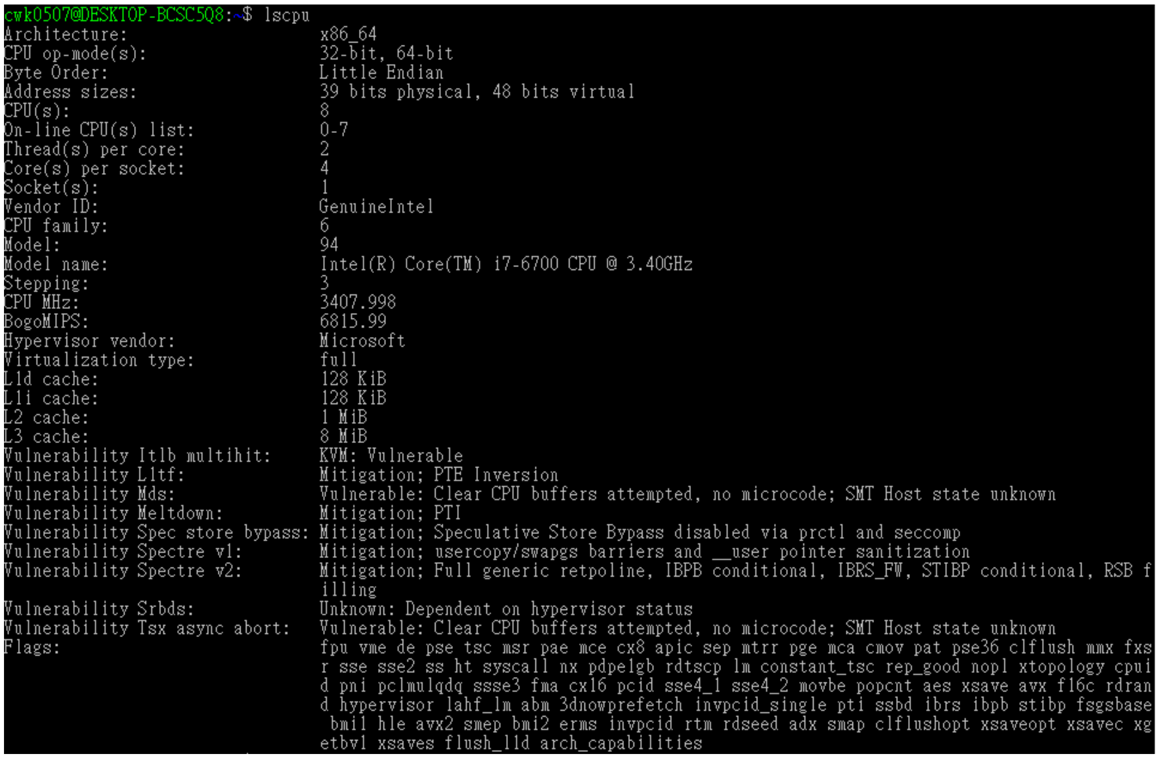
2. Use a few Linux commands to collect the hardware information of your computer to draw the memory hierarchy diagram (see, e.g., Slide 54 in Lecture 1). List the used commands and briefly explain what they are used for.

Several commands were used to collect information of different Level in the memory hierarchy diagram.

Level		Command used	Size in my PC
L0	Regs	lscpu – display information about the CPU architecture, which include information about the CPU cache	128 KiB
L1	L1 cache		128 KiB
L2	L2 cache		1 KiB
L3	L3 cache		8 MiB
L4	Main memory (DRAM)	free –h – display amount of free and used memory in the system in a human readable format, which includes the total installed memory	Mem: 12GiB Swap: 4 GiB
L5	Local secondary storage	df –h – report the file system disk space usage in a human readable format	251GB for /dev/sdb 1.9TB for C:\
L6	Remote secondary storage	N/A	N/A

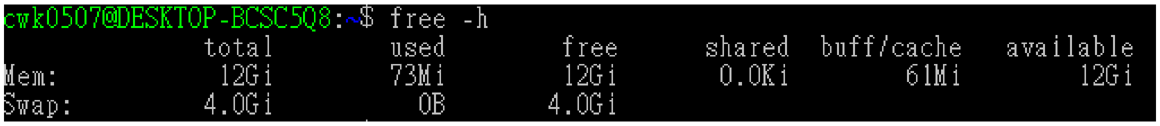
Screen shot:

lscpu:



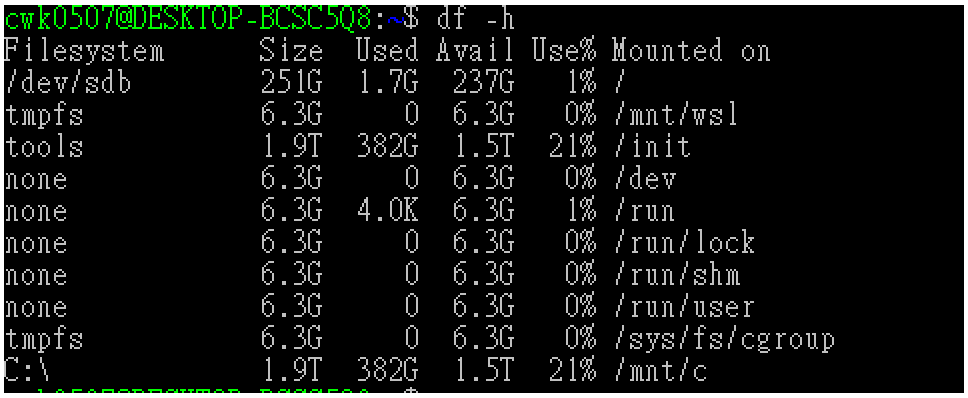
```
cwk0507@DESKTOP-BCSC5Q8:~$ lscpu
Architecture:                    x86_64
CPU op-mode(s):                  32-bit, 64-bit
Byte Order:                      Little Endian
Address sizes:                   39 bits physical, 48 bits virtual
CPU(s):                          8
On-line CPU(s) list:             0-7
Thread(s) per core:              2
Core(s) per socket:              4
Socket(s):                       1
Vendor ID:                       GenuineIntel
CPU family:                      6
Model:                           94
Model name:                      Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
Stepping:                        3
CPU MHz:                         3407.998
BogoMIPS:                        6815.99
Hypervisor vendor:               Microsoft
Virtualization type:             full
L1d cache:                       128 KiB
L1i cache:                       128 KiB
L2 cache:                        1 MiB
L3 cache:                        8 MiB
Vulnerability Itlb multihit:     KVM: Vulnerable
Vulnerability L1tf:              Mitigation; PTE Inversion
Vulnerability Mds:               Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
Vulnerability Meltdown:          Mitigation; PTI
Vulnerability Spec store bypass: Mitigation; Speculative Store Bypass disabled via prctl and seccomp
Vulnerability Spectre v1:        Mitigation; usercopy/swapgs barriers and __user pointer sanitization
Vulnerability Spectre v2:        Mitigation; Full generic retpoline, IBPB conditional, IBRS_FW, STIBP conditional, RSB f
                                 illing
Vulnerability Srbds:             Unknown: Dependent on hypervisor status
Vulnerability Tsx async abort:   Vulnerable: Clear CPU buffers attempted, no microcode; SMT Host state unknown
Flags:                           fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxs
                                 r sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtopology cpui
                                 d pni pclmulqdq ssse3 fma cx16 pcid sse4_1 sse4_2 movbe popcnt aes xsave avx f16c rdran
                                 d hypervisor lahf_lm abm 3dnowprefetch invpcid_single pti ssbd ibrs ibpb stibp fsgsbase
                                  bmi1 hle avx2 smep bmi2 erms invpcid rtm rdseed adx smap clflushopt xsaveopt xsavec xg
                                 etbv1 xsaves flush_l1d arch_capabilities
```

free –h



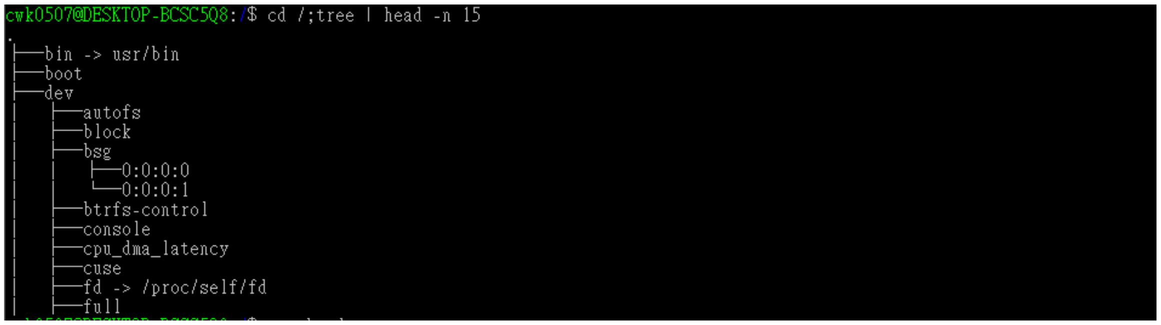
```
cwk0507@DESKTOP-BCSC5Q8:~$ free -h
              total        used        free      shared  buff/cache   available
Mem:           12Gi        73Mi        12Gi       0.0Ki        61Mi        12Gi
Swap:         4.0Gi          0B       4.0Gi
```

df –h



```
cwk0507@DESKTOP-BCSC5Q8:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb        251G  1.7G  237G   1% /
tmpfs           6.3G     0  6.3G   0% /mnt/wsl
tools           1.9T  382G  1.5T  21% /init
none            6.3G     0  6.3G   0% /dev
none            6.3G  4.0K  6.3G   1% /run
none            6.3G     0  6.3G   0% /run/lock
none            6.3G     0  6.3G   0% /run/shm
none            6.3G     0  6.3G   0% /run/user
tmpfs           6.3G     0  6.3G   0% /sys/fs/cgroup
C:\             1.9T  382G  1.5T  21% /mnt/c
```

3. Install the Linux “tree” command if your Linux system does not have it, e.g., sudo apt install tree. Run the commands
   cd /; tree | head -n 15
   Paste the output into your report and briefly explain what these commands did.



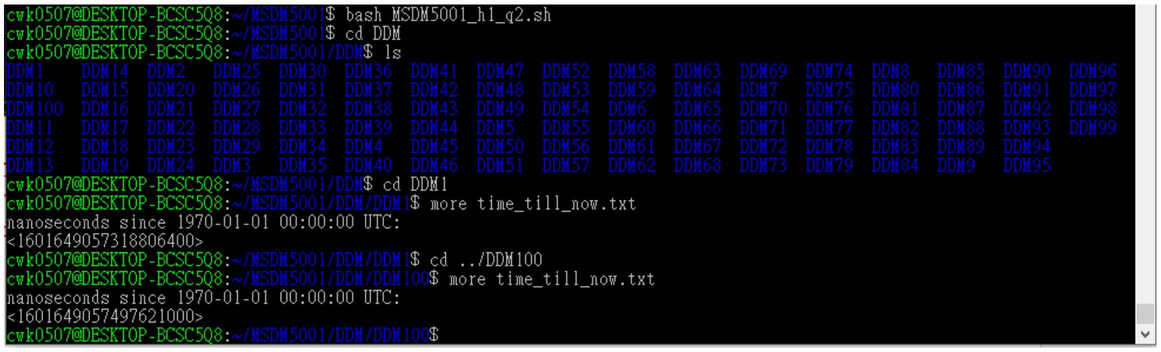
```
cwk0507@DESKTOP-BCSC5Q8:/$ cd /;tree | head -n 15
.
├──bin -> usr/bin
├──boot
├──dev
│   ├──autofs
│   ├──block
│   ├──bsg
│   │   ├──0:0:0:0
│   │   └──0:0:0:1
│   ├──btrfs-control
│   ├──console
│   ├──cpu_dma_latency
│   ├──cuse
│   ├──fd -> /proc/self/fd
│   ├──full
```

The command first change the current directory to the root; then display the first 15 lines of the output of command “tree” which listed contents of directories in a tree-like format.

2. Bash script

Write a bash script to create 100 directories/folders, whose names are “DDM1, DDM2, DDM3, ..., DDM100”. In each directory, generate a text file, “time till now.txt”, in which the content is nanoseconds since 1970-01-01 00:00:00 UTC:
< XXXXXXXXXXXXXXXXXXX >
The digits in <> should be calculated when you execute the script. (Hint: you may use the Linux command “date”. The same command in macOS/UNIX may not work.)

The code are in the file MSDM5001_h1_q2.sh which can be clone via “https://github.com/cwk0507/MSDM5001”. Some of the output are captured below.



```
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001$ bash MSDM5001_h1_q2.sh
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001$ cd DDM
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001/DDM$ ls
DDM1    DDM14  DDM2   DDM25  DDM30  DDM36  DDM41  DDM47  DDM52  DDM58  DDM63  DDM69  DDM74  DDM8   DDM85  DDM90  DDM96
DDM10   DDM15  DDM20  DDM26  DDM31  DDM37  DDM42  DDM48  DDM53  DDM59  DDM64  DDM7   DDM75  DDM80  DDM86  DDM91  DDM97
DDM100  DDM16  DDM21  DDM27  DDM32  DDM38  DDM43  DDM49  DDM54  DDM6   DDM65  DDM70  DDM76  DDM81  DDM87  DDM92  DDM98
DDM11   DDM17  DDM22  DDM28  DDM33  DDM39  DDM44  DDM5   DDM55  DDM60  DDM66  DDM71  DDM77  DDM82  DDM88  DDM93  DDM99
DDM12   DDM18  DDM23  DDM29  DDM34  DDM4   DDM45  DDM50  DDM56  DDM61  DDM67  DDM72  DDM78  DDM83  DDM89  DDM94
DDM13   DDM19  DDM24  DDM3   DDM35  DDM40  DDM46  DDM51  DDM57  DDM62  DDM68  DDM73  DDM79  DDM84  DDM9   DDM95
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001/DDM$ cd DDM1
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001/DDM/DDM1$ more time_till_now.txt
nanoseconds since 1970-01-01 00:00:00 UTC:
<1601649057318806400>
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001/DDM/DDM1$ cd ../DDM100
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001/DDM/DDM100$ more time_till_now.txt
nanoseconds since 1970-01-01 00:00:00 UTC:
<1601649057497621000>
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001/DDM/DDM100$
```

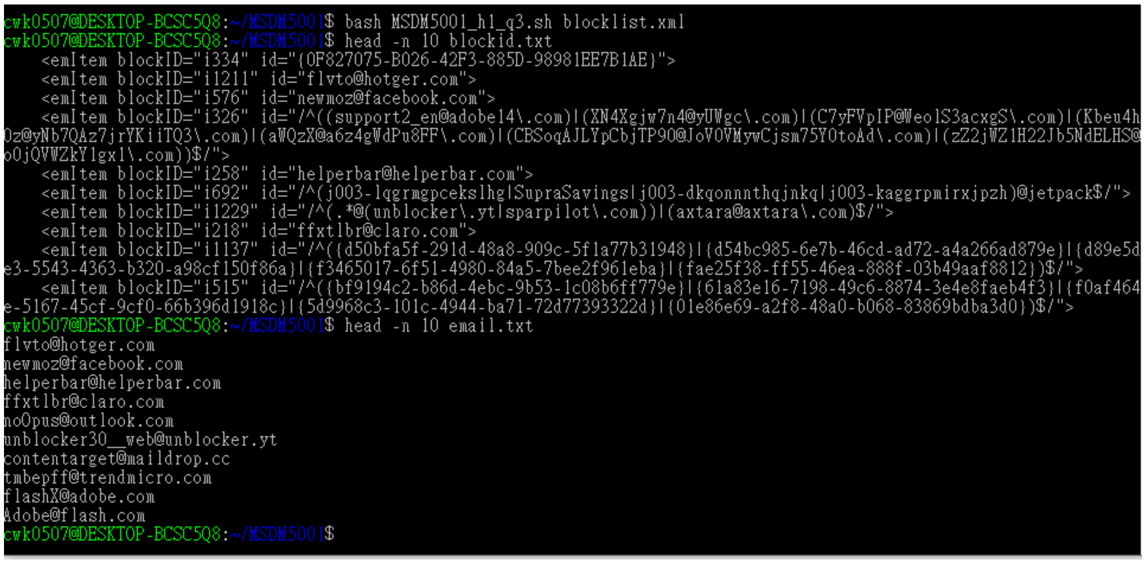
3. Regular expression

Write bash or python scripts to get the desired data from the “blocklist.xml” file. You should simply print the whole lines.

1. Print all the text lines with the “blockID” values that start with the letter “i” or “g”, and end with digits, e.g., “. (Tip: In the xml file, “blockID” is the attribute name and “i334” is the attribute value.)

2. Print all the text lines where the "ID" values are email addresses. Skip the email addresses that are written by regular expressions containing special characters, such as "\, /, ^ ".

The code are in the file MSDM5001_h1_q3.sh which can be clone via "https://github.com/cwk0507/MSDM5001". After running the code, the result of Q3.1 will be stored in blockid.txt and the result of Q3.2 will be stored in email.txt. The first 10 lines of both output files are captured below.



```
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001$ bash MSDM5001_h1_q3.sh blocklist.xml
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001$ head -n 10 blockid.txt
    <emItem blockID="i334" id="{0F827075-B026-42F3-885D-98981EE7B1AE}">
    <emItem blockID="i1211" id="flvto@hotger.com">
    <emItem blockID="i576" id="newmoz@facebook.com">
    <emItem blockID="i326" id="/^((support2_en@adobe14\.com)|(XN4Xgjw7n4@yUWgc\.com)|(C7yFVpIP@WeolS3acxgS\.com)|(Kbeu4h
0z@yNb7QAz7jrYKiiTQ3\.com)|(aWQzX@a6z4gWdPu8FF\.com)|(CBSoqAJLYpCbjTP90@JoV0VMywCjsm75Y0toAd\.com)|(zZ2jWZ1H22Jb5NdELHS@
o0jQVWZkY1gx1\.com))$/">
    <emItem blockID="i258" id="helperbar@helperbar.com">
    <emItem blockID="i692" id="/^(j003-lqgrmgpcekslhg|SupraSavings|j003-dkqonnnthqjnkq|j003-kaggrpmirxjpzh)@jetpack$/">
    <emItem blockID="i1229" id="/^(.*@(unblocker\.yt|sparpilot\.com))|(axtara@axtara\.com)$/">
    <emItem blockID="i218" id="ffxtlbr@claro.com">
    <emItem blockID="i1137" id="/^({d50bfa5f-291d-48a8-909c-5f1a77b31948}|{d54bc985-6e7b-46cd-ad72-a4a266ad879e}|{d89e5d
e3-5543-4363-b320-a98cf150f86a}|{f3465017-6f51-4980-84a5-7bee2f961eba}|{fae25f38-ff55-46ea-888f-03b49aaf8812})$/">
    <emItem blockID="i515" id="/^({bf9194c2-b86d-4ebc-9b53-1c08b6ff779e}|{61a83e16-7198-49c6-8874-3e4e8faeb4f3}|{f0af464
e-5167-45cf-9cf0-66b396d1918c}|{5d9968c3-101c-4944-ba71-72d77393322d}|{01e86e69-a2f8-48a0-b068-83869bdba3d0})$/">
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001$ head -n 10 email.txt
flvto@hotger.com
newmoz@facebook.com
helperbar@helperbar.com
ffxtlbr@claro.com
noOpus@outlook.com
unblocker30__web@unblocker.yt
contentarget@maildrop.cc
tmbepff@trendmicro.com
flashX@adobe.com
Adobe@flash.com
cwk0507@DESKTOP-BCSC5Q8:~/MSDM5001$
```