

# 5001 Homework 1

ZHANG Juntao  
20908272

September 30, 2022

## 1 Problem 1: Linux operating system and memory hierarchy

Use Gitpod, which provide a cloud-based Linux container. The version of bash is 5.0.17(1), and the detail is as Figure 1 shows.

```
gitpod /workspace/KlausZhangjt.github.io (main) $ echo $SHELL
/bin/bash
gitpod /workspace/KlausZhangjt.github.io (main) $ bash --version
GNU bash, version 5.0.17(1)-release (x86_64-pc-linux-gnu)
Copyright (C) 2019 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>

This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
```

Figure 1: Details of bash version

### 1.1 Question 1

Run the command "top" on the terminal to real-time display the resource occupancy status of each process in the system, and save a screenshot as Figure 2.

```
top - 05:43:35 up 4:25, 0 users, load average: 6.31, 4.37, 4.45
Tasks: 17 total, 1 running, 16 sleeping, 0 stopped, 0 zombie
%Cpu(s): 9.7 us, 2.1 sy, 0.0 ni, 87.5 id, 0.0 wa, 0.0 hi, 0.7 si, 0.0 st
MiB Mem : 64310.7 total, 1038.6 free, 28588.8 used, 34683.3 buff/cache
MiB Swap: 384000.0 total, 383966.2 free, 33.8 used, 34965.3 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
34	root	10	-10	735136	30760	16492	S	1.0	0.0	0:08.41	supervisor
242	gitpod	20	0	13.8g	1.8g	350984	S	0.7	2.8	0:59.72	ld-linux-x86-64
1	root	10	-10	732576	18252	13772	S	0.0	0.0	0:00.05	supervisor
61	gitpod	10	-10	2620	328	256	S	0.0	0.0	0:00.12	sh
62	gitpod	20	0	12424	9128	3384	S	0.0	0.0	0:00.13	bash
63	gitpod	20	0	721216	10176	8324	S	0.0	0.0	0:00.00	status
75	gitpod	20	0	2616	324	256	S	0.0	0.0	0:00.00	remote-dev-serv
84	gitpod	20	0	2616	1500	1380	S	0.0	0.0	0:00.00	launcher.sh
909	gitpod	15	-5	939052	86064	35260	S	0.0	0.1	0:10.66	node
928	gitpod	15	-5	635748	47992	32504	S	0.0	0.1	0:00.78	node
1212	gitpod	20	0	10604	1648	1308	S	0.0	0.0	0:00.11	fsnotifier
1249	gitpod	15	-5	981228	112744	37264	S	0.0	0.2	0:04.12	node
1262	gitpod	15	-5	704440	36944	28512	S	0.0	0.1	0:00.17	node
1300	gitpod	15	-5	637416	85280	30244	S	0.0	0.1	0:00.58	node
1337	gitpod	15	-5	12408	8956	3268	S	0.0	0.0	0:00.08	bash
1604	gitpod	15	-5	589880	39032	28236	S	0.0	0.1	0:00.16	node
2202	gitpod	20	0	9076	3700	3160	R	0.0	0.0	0:00.02	top

Figure 2: Result of "top"

### 1.2 Question 2

Use Linux commands to collect the hardware information.

Command list:

1. `lscpu` : check the information of CPU
2. `free -h` : display memory usage information briefly
3. `cat /proc/meminfo` : display RAM usage information in detail
4. `getconf -a — grep CACHE` : check the details of L1 cache, L2 cache and L3 cache.
5. (1) `sudo apt-get install lshw` (2) `sudo lshw -class memory` : get the information of main memory
6. `df -h` : check the information of disk storage

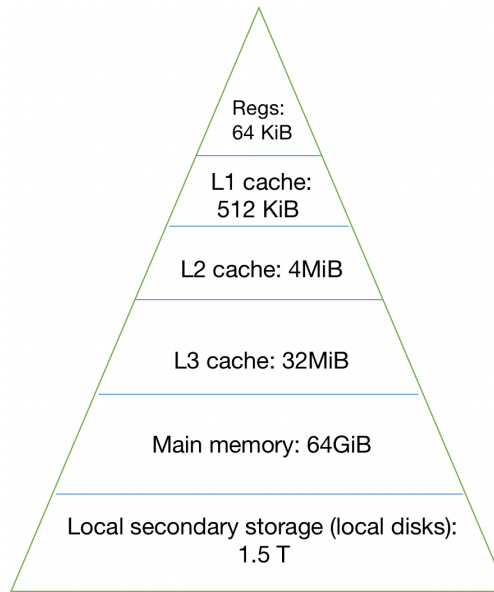


Figure 3: Memory hierarchy diagram

```
gitpod /workspace/KlausZhangjt.github.io (main) $ free -h
              total        used        free      shared  buff/cache   available
Mem:           62Gi         19Gi         13Gi          62Mi        30Gi         43Gi
Swap:          374Gi         248Mi         374Gi
gitpod /workspace/KlausZhangjt.github.io (main) $ cat /proc/meminfo
MemTotal:        65854172 kB
MemFree:         12545592 kB
MemAvailable:    45314804 kB
Buffers:          1420 kB
Cached:          30051692 kB
SwapCached:       12424 kB
Active:           6424176 kB
Inactive:         39565376 kB
Active(anon):     1291144 kB
Inactive(anon):   14486968 kB
Active(file):     5133032 kB
Inactive(file):   25078408 kB
Unevictable:       82692 kB
Mlocked:          82692 kB
SwapTotal:       393215996 kB
SwapFree:        392961276 kB
Dirty:           1599204 kB
Writeback:         0 kB
```

Figure 4: Result of command 2-3

```

gitpod /workspace/KlausZhangjt.github.io (main) $ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:             Little Endian
Address sizes:          48 bits physical, 48 bits virtual
CPU(s):                 16
On-line CPU(s) list:   0-15
Thread(s) per core:     2
Core(s) per socket:     8
Socket(s):              1
NUMA node(s):          1
Vendor ID:              AuthenticAMD
CPU family:             25
Model:                  1
Model name:             AMD EPYC 7B13
Stepping:                0
CPU MHz:                2450.000
BogoMIPS:               4900.00
Hypervisor vendor:      KVM
Virtualization type:    full
L1d cache:              256 KiB
L1i cache:              256 KiB
L2 cache:               4 MiB
L3 cache:               32 MiB
NUMA node0 CPU(s):      0-15
Vulnerability Itlb multihit: Not affected
Vulnerability L1tf:      Not affected
Vulnerability Mds:       Not affected
Vulnerability Meltdown:  Not affected
Vulnerability Mmio stale data: Not affected
Vulnerability Retbleed:  Not affected
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: Retpolines, IBPB conditional, IBRS_FW, STIBP conditional, RSB filling
Vulnerability Srbds:      Not affected
Vulnerability Tsx async abort: Not affected
Flags:                    fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr
                           r sse sse2 ht syscall nx mmxext fxsr_opt pdpe1gb rdtscp lm constant_tsc rep_good nopl n
                           ontop tsc cpuid extd_apicid tsc_known_freq pni pclmulqdq ssse3 fma cx16 pcid sse4_1 ss
                           e4_2 movbe popcnt aes xsave avx f16c rdrand hypervisor lahf_lm cmp_legacy cr8_legacy ab
                           m sse4b misalignsse 3dnowprefetch osvw topoptel invpcid_single ssbd ibrs lbpb stibp vmm
                           all fsgsbase tsc_adjust bmi1 avx2 smep bmi2 erms invpcid rdseed adx snap clflushopt clw
                           b sha_ni xsaveopt xsavec xgetbv1 clzero xsaveprtr arat nrip_save umip rdpid fsrm

```

Figure 5: Result of command 1

```

gitpod /workspace/KlausZhangjt.github.io (main) $ sudo lshw -class memory
*-memory
   description: System memory
   physical id: 0
   size: 64GiB
gitpod /workspace/KlausZhangjt.github.io (main) $ getconf -a | grep CACHE
LEVEL1_ICACHE_SIZE      32768
LEVEL1_ICACHE_ASSOC     8
LEVEL1_ICACHE_LINESIZE  64
LEVEL1_DCACHE_SIZE      32768
LEVEL1_DCACHE_ASSOC     8
LEVEL1_DCACHE_LINESIZE  64
LEVEL2_CACHE_SIZE       524288
LEVEL2_CACHE_ASSOC      8
LEVEL2_CACHE_LINESIZE   64
LEVEL3_CACHE_SIZE       268435456
LEVEL3_CACHE_ASSOC      0
LEVEL3_CACHE_LINESIZE   64
LEVEL4_CACHE_SIZE       0
LEVEL4_CACHE_ASSOC      0
LEVEL4_CACHE_LINESIZE   0
gitpod /workspace/KlausZhangjt.github.io (main) $ df -h
Filesystem      Size  Used Avail Use% Mounted on
/.workspace/mark 1.5T  501G  999G  34% /
tmpfs           64M   0   64M   0% /dev
/dev/sda1       512G  17G  496G   4% /dev/termination-log
shm             64M   0   64M   0% /dev/shm
tmpfs          32G   0   32G   0% /sys/firmware
/dev/md42       30G   63M   30G   1% /workspace
/dev/md44       1.5T  501G  999G  34% /etc/hostname
tmpfs          32G   23M   32G   1% /tmp
tmpfs          32G   0   32G   0% /proc/acpi
tmpfs          64M   0   64M   0% /proc/keys
tmpfs          32G   0   32G   0% /proc/scsi

```

Figure 6: Result of command 4-6

### 1.3 Question 3

Command:

(1) sudo apt install tree (2) cd /etc; tree — head -n 30

These commands are to list the first 30 files of the etc directory as a treemap, and the output is as Figure 7 shows.

```

gitpod /workspace/KlausZhangjt.github.io (main) $ cd /etc; tree | head -n 30
├── adduser.conf
├── alternatives
│   ├── aclocal -> /usr/bin/aclocal-1.16
│   ├── aclocal.1.gz -> /usr/share/man/man1/aclocal-1.16.1.gz
│   ├── animate -> /usr/bin/animate-im6.q16
│   ├── animate.1.gz -> /usr/share/man/man1/animate-im6.q16.1.gz
│   ├── animate-im6 -> /usr/bin/animate-im6.q16
│   ├── animate-im6.1.gz -> /usr/share/man/man1/animate-im6.q16.1.gz
│   ├── arptables -> /usr/sbin/arptables-nft
│   ├── arptables-restore -> /usr/sbin/arptables-nft-restore
│   ├── arptables-save -> /usr/sbin/arptables-nft-save
│   ├── automake -> /usr/bin/automake-1.16
│   ├── automake.1.gz -> /usr/share/man/man1/automake-1.16.1.gz
│   ├── awk -> /usr/bin/gawk
│   ├── awk.1.gz -> /usr/share/man/man1/gawk.1.gz
│   ├── builtins.7.gz -> /usr/share/man/man7/bash-builtins.7.gz
│   ├── bzip -> /usr/bin/bzip
│   ├── bzip.1.gz -> /usr/share/man/man1/bzip.1.gz
│   ├── c++ -> /usr/bin/g++
│   ├── c++.1.gz -> /usr/share/man/man1/g++.1.gz
│   ├── c89 -> /usr/bin/c89-gcc
│   ├── c89.1.gz -> /usr/share/man/man1/c89-gcc.1.gz
│   ├── c99 -> /usr/bin/c99-gcc
│   ├── c99.1.gz -> /usr/share/man/man1/c99-gcc.1.gz
│   ├── cc -> /usr/bin/gcc
│   ├── cc.1.gz -> /usr/share/man/man1/gcc.1.gz
│   ├── compare -> /usr/bin/compare-im6.q16
│   ├── compare.1.gz -> /usr/share/man/man1/compare-im6.q16.1.gz
│   ├── compare-im6 -> /usr/bin/compare-im6.q16

```

Figure 7: Result of question 3

## 2 Problem 2: Bash script

Bash script(codes of q2.sh):

```
1 #!/bin/bash
2
3 for i in $(seq 1 180)
4 do
5     echo "make directory postproc$i"
6     mkdir MSDM$i
7     touch /workspace/KlausZhangjt.github.io/MSDM$i/"time till now.txt"
8     echo microseconds since 1970-01-01 00:00:00 UTC: >> \
9     ./MSDM$i/"time till now.txt"
10    cur_ns='date +%s%N'
11    echo "scale=0; $cur_ns/1000" | bc >> ./MSDM$i/"time till now.txt"
12    #remian integer part
13 done
```

Then use these commands:

(1) `chmod +x ./q1.sh`

(2) `./q1.sh`

Then we can get 180 folders, whose names are "MSDM1, MSDM2, MSDM3, ..., MSDM180". And in each folder, there is a text file "time till now.txt", in which the content is the microseconds from 1970-01-01 00:00:00 UTC to now.

Besides, we only remain the integer part the microseconds.

Part of the results is shown in Figure 8.

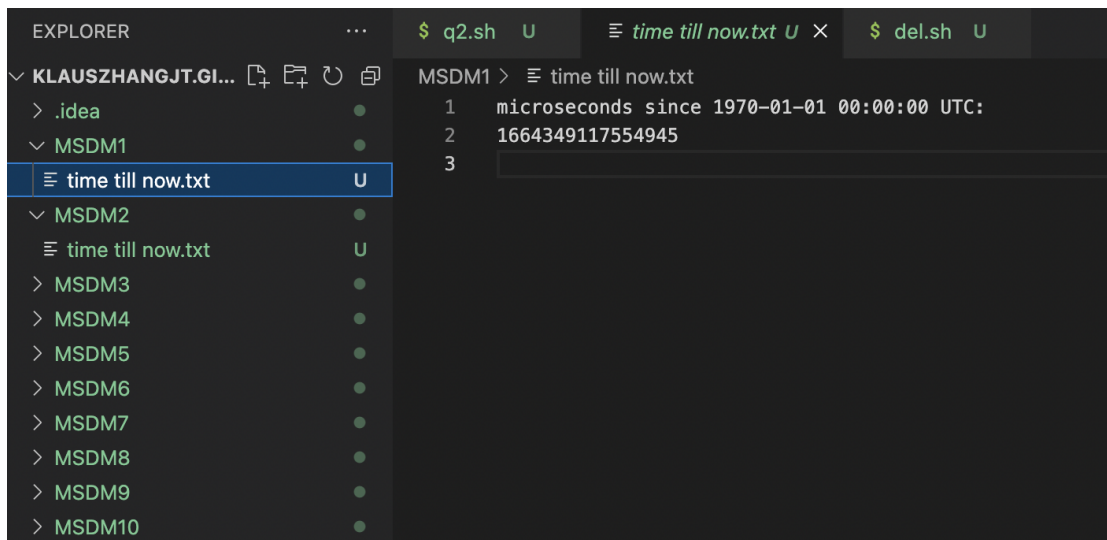


Figure 8: Part result of problem 2

### 3 Problem 3: Regular expression

Python script(codes of q3.py), version: python3.9

```
1 import re
2
3 #question 1
4 f = open("blocklist.xml", encoding="utf-8")
5 bl = f.read()
6 regular1 = r'\<emItem\sblockID=\"[i|d].*[0-9]\"\\s.*>'
7 pattern1 = re.compile(regular1)
8 result1 = pattern1.findall(bl)
9 for i in range(len(result1)):
10     print(result1[i])
11 print('There are', len(result1), 'text lines satisfy the requirement in
    question 1.')
12
13 #question 2
14 f = open("blocklist.xml", encoding="utf-8")
15 bl = f.read()
16 regular2 = r'<.*\sid=\"[^\s]*[c|o][o|r][m|g]\".*>'
17 pattern2 = re.compile(regular2)
18 result2 = pattern2.findall(bl)
19 for i in range(len(result2)):
20     print(result2[i])
21 print('There are', len(result2), 'text lines satisfy the requirement in
    question 2.')
```

And then we can get the results in Figure 9 and Figure 10:

There are 455 text lines that satisfy the requirement in question 1;

And there are 163 text lines that satisfy the requirement in question 2.

```
<emItem blockID="i13" id="{E8E88AB0-7182-11DF-904E-6045E0D72085}">
<emItem blockID="i48" id="admin@youtubespeedup.com">
<emItem blockID="i47" id="youtube@youtube2.com">
<emItem blockID="i10" id="{8CE11043-9A15-4207-A565-0C94C42D590D}">
<emItem blockID="i43" id="supportaccessplugin@gmail.com">
<emItem blockID="i7" id="{2224e955-00e9-4613-a844-ce69fccaae91}">
<emItem blockID="i21" id="support@update-firefox.com">
<emItem blockID="i1492" id="googlotim@gmail.com">
<emItem blockID="i1493" id="{de71f09a-3342-48c5-95c1-4b0f17567554}">
<emItem blockID="i1522" id="/^(ciscowebxstart1@cisco\.com|ciscowebxstart_test@cisco\.com|ciscowebxstart@cisco\.com|ciscowebxgpc@cisco\.com)$/"/>
<emItem blockID="i1523" id="{a0d7ccb3-214d-498b-b4aa-0e8fda9a7bf7}">
<emItem blockID="i1524" id="ext@alibonus.com">
<emItem blockID="d33f6d48-a555-49dd-96ff-8d75473403a8" id="mozilla_cc2@internetdownloadmanager.com">
<emItem blockID="d6425f24-8c9e-4c0a-89b4-6890fc68d5c9" id="/^\\{(9321F452-96D5-11E6-BC3E-3769C7AD2208)|\\{(18ED1ECA-96D3-11E6-A373-BD66C7AD2208)}\\}$/"/>
There are 455 text lines satisfy the requirement in question 1.
```

Figure 9: Result of question 1

```

<emItem blockID="i60" id="youthb3@youthb3.com">
<emItem blockID="i90" id="videoplugin@player.com">
<emItem blockID="i48" id="admin@youtubespeedup.com">
<emItem blockID="i47" id="youtube@youtube2.com">
<emItem blockID="i43" id="supportaccessplugin@gmail.com">
<emItem blockID="i21" id="support@update-firefox.com">
<emItem blockID="i1492" id="googlotim@gmail.com">
<emItem blockID="i1524" id="ext@alibonus.com">
<emItem blockID="89a61123-79a2-45d1-aec2-97afca0863eb" id="InternetProtection@360safe.com">
<emItem blockID="d33f6d48-a555-49dd-96ff-8d75473403a8" id="mozilla_cc2@internetdownloadmanager.com">
<emItem blockID="e16408c3-4e08-47fd-85a9-3cbbce534e95" id="WebProtection@360safe.com">
<emItem blockID="28736359-700e-4b61-9c50-0b533a6bac55" id="xdict@www.iciba.com">
<emItem blockID="22431713-a93b-40f4-8264-0b341b5f6454" id="fi@dictionaries.addons.mozilla.org">
<emItem blockID="baf7f735-d6b6-410a-8cc8-25c60f7c57e2" id="adbeaver@adbeaver.org">
There are 163 text lines satisfy the requirement in question 2.

```

Figure 10: Result of question 2

## 4 Problem 4: GitHub webpage

Create a webpage on GitHub.com using the “index.html” file. The domain name is:

<https://klauszhangjt.github.io/>