ECE 4310
Operating Systems for Embedded Application

Project 2

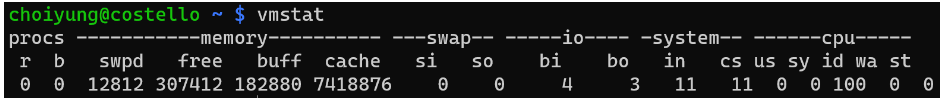
Choi Tim Antony Yung

April 15, 2021

1 vmstat and free

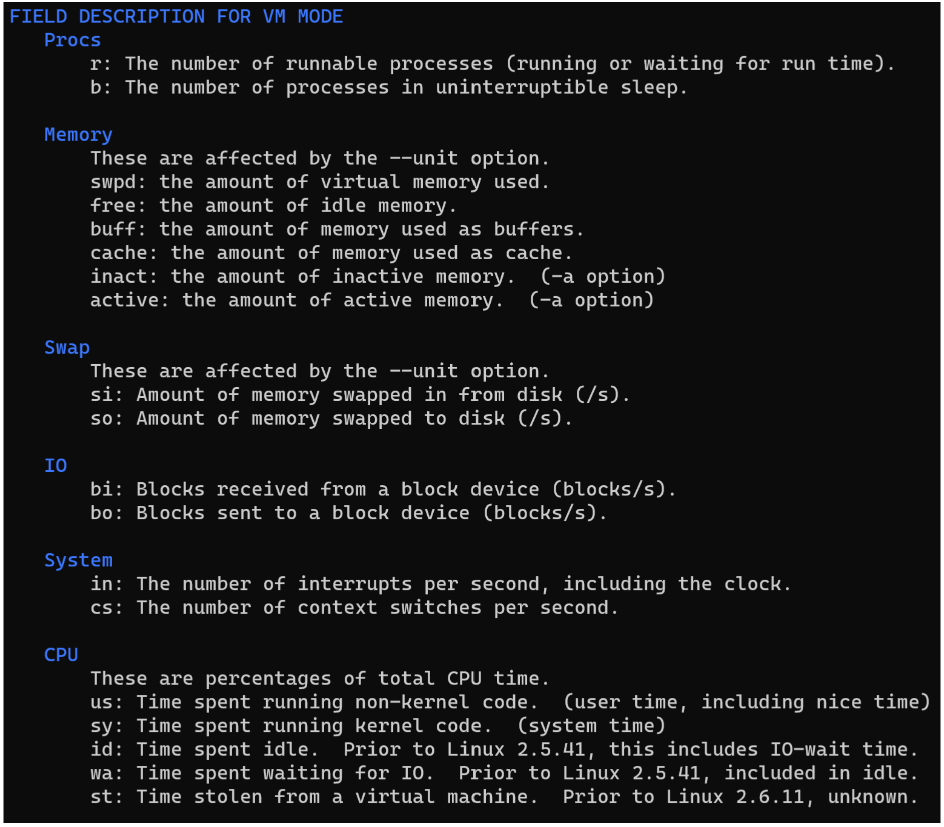
1.1 vmstat

Figure 1: Output of `vmstat`



```
choiyung@costello ~ $ vmstat
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st
 0  0  12812 307412 182880 7418876    0    0     4     3   11   11  0  0 100  0  0
```

Figure 2: Output of `man vmstat`



```
FIELD DESCRIPTION FOR VM MODE
   Procs
       r: The number of runnable processes (running or waiting for run time).
       b: The number of processes in uninterruptible sleep.

   Memory
       These are affected by the --unit option.
       swpd: the amount of virtual memory used.
       free: the amount of idle memory.
       buff: the amount of memory used as buffers.
       cache: the amount of memory used as cache.
       inact: the amount of inactive memory.  (-a option)
       active: the amount of active memory.  (-a option)

   Swap
       These are affected by the --unit option.
       si: Amount of memory swapped in from disk (/s).
       so: Amount of memory swapped to disk (/s).

   IO
       bi: Blocks received from a block device (blocks/s).
       bo: Blocks sent to a block device (blocks/s).

   System
       in: The number of interrupts per second, including the clock.
       cs: The number of context switches per second.

   CPU
       These are percentages of total CPU time.
       us: Time spent running non-kernel code.  (user time, including nice time)
       sy: Time spent running kernel code.  (system time)
       id: Time spent idle.  Prior to Linux 2.5.41, this includes IO-wait time.
       wa: Time spent waiting for IO.  Prior to Linux 2.5.41, included in idle.
       st: Time stolen from a virtual machine.  Prior to Linux 2.6.11, unknown.
```

According to the `man` page:

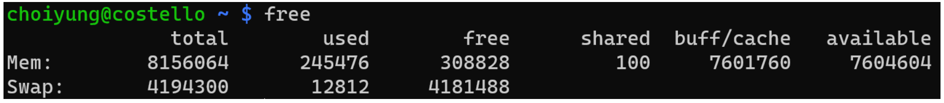
- r indicate there are currently no processes running or waiting;
- b indicate there are currently no processes in uninterruptible sleep;
- swpd is the amount of virtual memory/swap memory/pagefile/backing store used;
- free is the amount of physical memory not in use;
- buff is the amount of physical memory used as buffers;
- cache is the amount of physical memory used as caches;
- si is the amount of swapping/paging in per second, indicating if some page was swapped back to physical memory;

- so is the amount of swapping/paging out per second, indicating if some page was swapped to disk to free frames;
- bi is the amount of blocks received from a block device, i.e. input from some hardware devices
- bo is the amount of blocks sent to a block device, i.e. output to some hardware devices
- in is the number of interrupts per second
- cs is the number of context switches per second
- us is the percentage of CPU time running non-kernel code
- sy is the percentage of CPU time running kernel code
- id is the percentage of CPU time being idle
- wa is the percentage of CPU time waiting for IO
- st is the percentage of CPU time stolen from virtual machine

From the `vmstat` output, it can be determined that no user space processes are currently running or sleeping; 12812B of backing store is in use; 307412B of physical memory is not in use; 182880B of physical memory is used as buffer; 7418876B of physical memory is used as cache; There are no paging in or paging out, meaning there are enough physical memory for processes running; Some blocks are being transferred indicating accesses to block devices (usually mass storage); Multiple processes are running, hence the interrupts and context switching; and CPU spent almost all its time being idle.

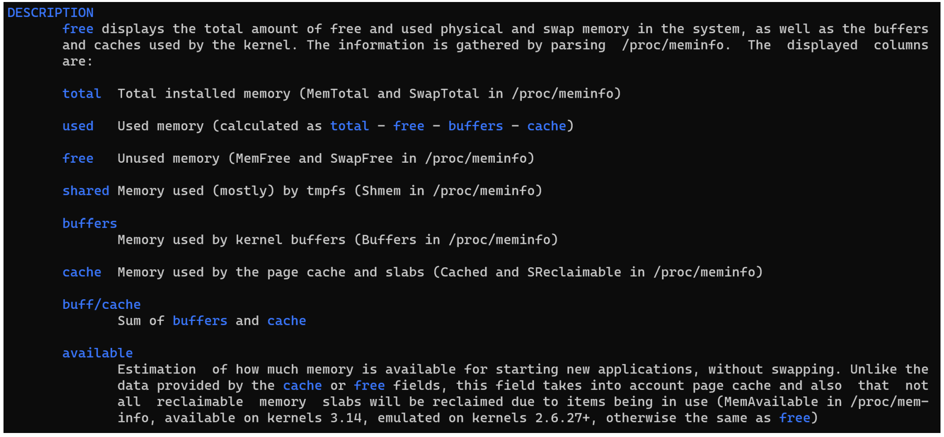
1.2 `free`

Figure 3: Output of `free`



```
choiyung@costello ~ $ free
               total        used        free      shared  buff/cache   available
Mem:         8156064      245476      308828         100     7601760     7604604
Swap:        4194300       12812     4181488
```

Figure 4: Output of `man free`



```
DESCRIPTION
       free displays the total amount of free and used physical and swap memory in the system, as well as the buffers
       and caches used by the kernel. The information is gathered by parsing  /proc/meminfo.  The  displayed  columns
       are:

       total  Total installed memory (MemTotal and SwapTotal in /proc/meminfo)

       used   Used memory (calculated as total - free - buffers - cache)

       free   Unused memory (MemFree and SwapFree in /proc/meminfo)

       shared Memory used (mostly) by tmpfs (Shmem in /proc/meminfo)

       buffers
              Memory used by kernel buffers (Buffers in /proc/meminfo)

       cache  Memory used by the page cache and slabs (Cached and SReclaimable in /proc/meminfo)

       buff/cache
              Sum of buffers and cache

       available
              Estimation  of how much memory is available for starting new applications, without swapping. Unlike the
              data provided by the cache or free fields, this field takes into account page cache and also  that  not
              all  reclaimable  memory  slabs will be reclaimed due to items being in use (MemAvailable in /proc/mem‐
              info, available on kernels 3.14, emulated on kernels 2.6.27+, otherwise the same as free)
```

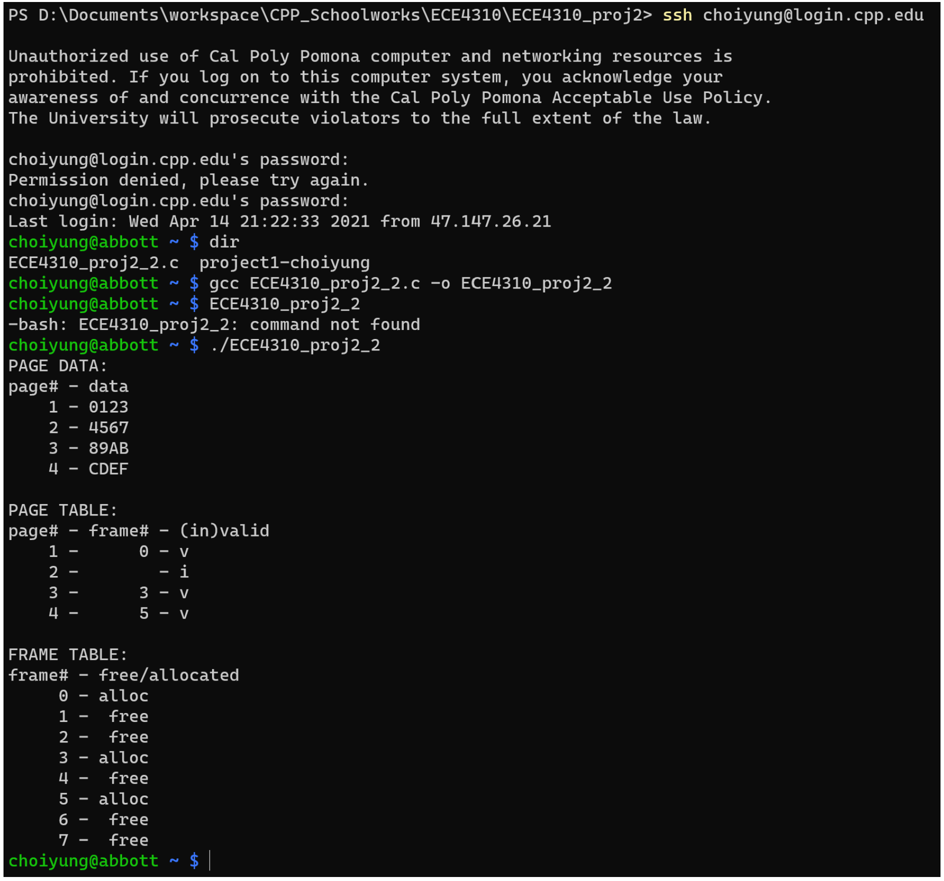
From the `free` output, it can be determined that the system have a total of 8156064B of physical memory, 245476B are allocated to processes, 308828B of physical memory are not used, 100B seems to be used mostly by tmpfs to provide a fast access file system, 7601760B are used as buffers or cache, and 7604604B can be allocated to new process. Also, there are a total of 4194300B of mass storage space allocated for backing store, 12812B of them are used and 4181488B are not.

2 Paging Simulation

Code

```c
#include <stdio.h>
#include <string.h>

void printtable(unsigned char phymem,unsigned char logmem, unsigned char pagesize, unsigned char* mapping, unsigned char* data)
{
    unsigned char totalpages = logmem/pagesize;
    unsigned char totalframes = phymem/pagesize;
    printf("PAGE DATA:\n");
    printf("page# - data\n");
    for (int i = 0; i < totalpages;i++)
    {
        printf("%5d - ", i+1);
        for (int j = 0; j < pagesize; j++)
        {
            printf("%02X", data[2*i+j]);
        }
        printf("\n");
    }

    printf("\n");
    printf("PAGE TABLE:\n");
    printf("page# - frame# - (in)valid\n");
    for (int i = 0; i < totalpages;i++)
    {
        printf("%5d - ", i+1);
        char frame = 0;
        char valid = 0;
        for (int j = 0; j < totalframes; j++)
        {
            if (mapping[j] == i+1)
            {
                frame = j;
                valid = 1;
                break;
            }
        }
        if (valid)
        {
            printf("%6d - v\n", frame);
        }
        else
        {
            printf("       - i\n");
        }
    }

    printf("\n");
    printf("FRAME TABLE:\n");
    printf("frame# - free/allocated\n");
    for (int j = 0; j < totalframes; j++)
    {
        printf("%6d - ", j);
        if (mapping[j] == 0)
        {
            printf(" free\n");
        }
        else
        {
            printf("alloc\n");
        }
    }
    return;
}


int main(int argc, char *argv[])
{
    unsigned char phymem = 16;
    unsigned char logmem = 8;
    unsigned char pagesize = 2;
    unsigned char mapping_arr[8] = {1,0,0,3,0,4,0,0};
    unsigned char *mapping = mapping_arr;
    unsigned char data_arr[8] = {0x01, 0x23, 0x45, 0x67, 0x89, 0xab, 0xcd, 0xef};
    unsigned char *data = data_arr;
    printtable(phymem,logmem,pagesize,mapping,data);
    return 0;
}
```

Figure 5: Output of the code



```
PS D:\Documents\workspace\CPP_Schoolworks\ECE4310\ECE4310_proj2> ssh choiyung@login.cpp.edu

Unauthorized use of Cal Poly Pomona computer and networking resources is
prohibited. If you log on to this computer system, you acknowledge your
awareness of and concurrence with the Cal Poly Pomona Acceptable Use Policy.
The University will prosecute violators to the full extent of the law.

choiyung@login.cpp.edu's password:
Permission denied, please try again.
choiyung@login.cpp.edu's password:
Last login: Wed Apr 14 21:22:33 2021 from 47.147.26.21
choiyung@abbott ~ $ dir
ECE4310_proj2_2.c  project1-choiyung
choiyung@abbott ~ $ gcc ECE4310_proj2_2.c -o ECE4310_proj2_2
choiyung@abbott ~ $ ECE4310_proj2_2
-bash: ECE4310_proj2_2: command not found
choiyung@abbott ~ $ ./ECE4310_proj2_2
PAGE DATA:
page# - data
    1 - 0123
    2 - 4567
    3 - 89AB
    4 - CDEF

PAGE TABLE:
page# - frame# - (in)valid
    1 -      0 - v
    2 -        - i
    3 -      3 - v
    4 -      5 - v

FRAME TABLE:
frame# - free/allocated
     0 - alloc
     1 -  free
     2 -  free
     3 - alloc
     4 -  free
     5 - alloc
     6 -  free
     7 -  free
choiyung@abbott ~ $
```