**Jurnal Praktikum**

**Sistem Operasi (CII-2H3)**

**Modul 6: Memori**

**Catatan**

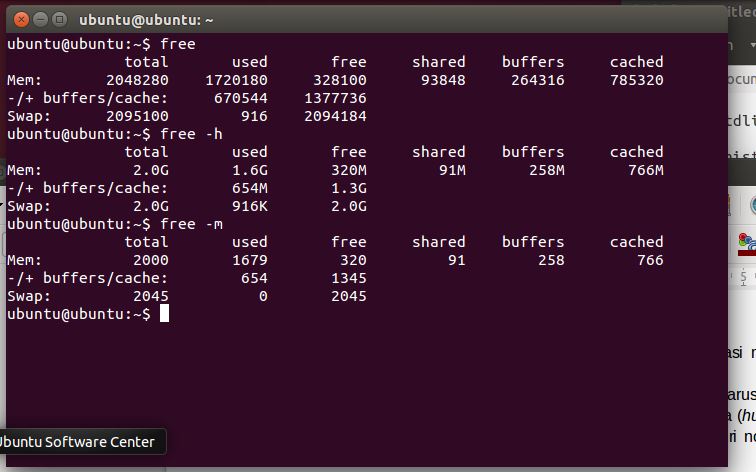
1. Praktikan wajib untuk screenshot setiap langkah yang dikerjakan hingga tampilan output akhir
2. Untuk soal source code, kumpulkan SS-nya saja
3. Praktikan wajib untuk melakukan screenshot lengkap dengan nama root. Contoh : root@username
4. Berikan identitas nama - nim dalam bentuk comment di Source Code
5. Harap kerjakan secara mandiri, jika tidak paham silahkan bertanya kepada Asisten Praktikum masing-masing. Dilarang mengcopy jawaban dan source code dari teman!

**Tujuan**

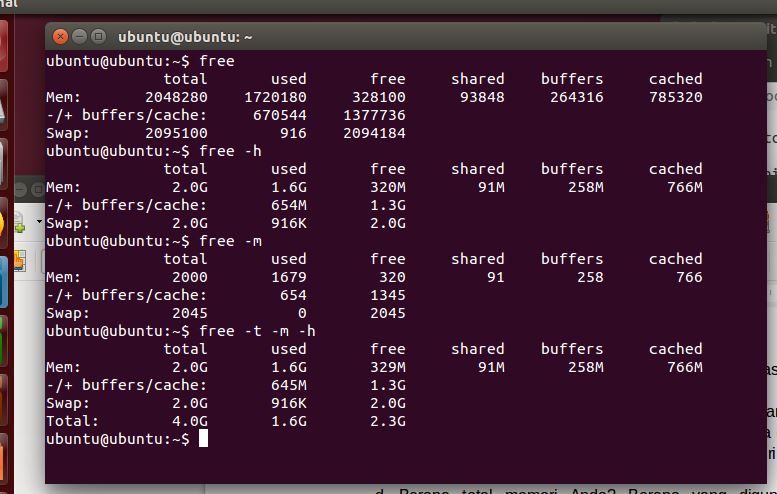
1. Mahasiswa mampu melakukan manajemen memori di linux
2. Mahasiswa mengetahui mekanisme translasi alamat memori

**Jurnal Praktikum**

1. Manajemen Memori
2. Terdapat beberapa tool untuk mengetahui informasi mengenai memori. Tool tersebut antara lain: free, meminfo dan vmstat
3. Baca manual perintah “free”. Option apakah yang harus diberikan jika kita ingin “free” menghasilkan output yang mudah dibaca (*human readable*)?
4. Jalankan perintah “free” disertai dengan option dari nomor sebelumnya! Ambillah screenshot hasilnya!

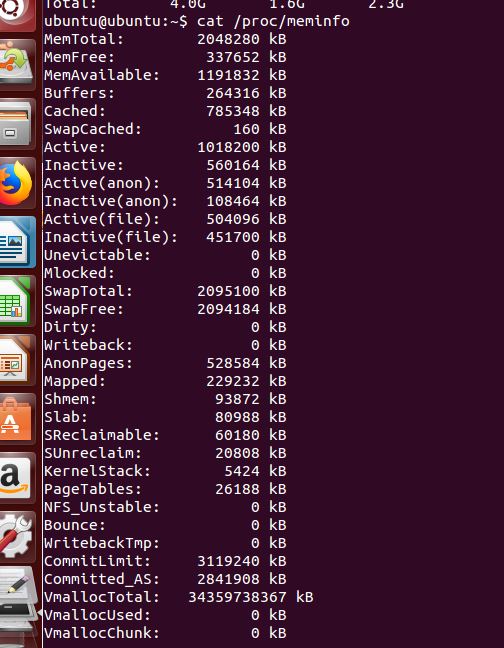


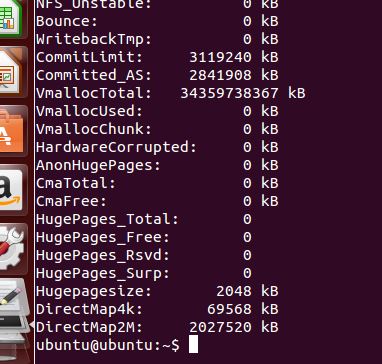
1. Berapa total memori Anda? Berapa yang digunakan? Berapa sisa memori?

* 

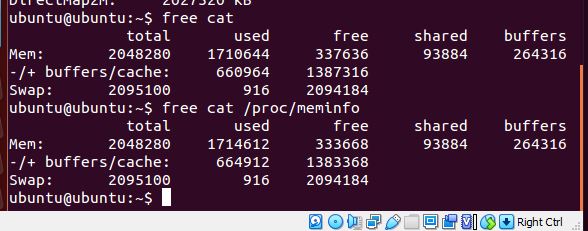
Total 4.0G Used 1.6G Free 2.3G

1. Jalankan perintah “cat /proc/meminfo”. Screenshot hasilnya!





1. Berapa total memori Anda? Berapa yang digunakan? Berapa sisa memori?

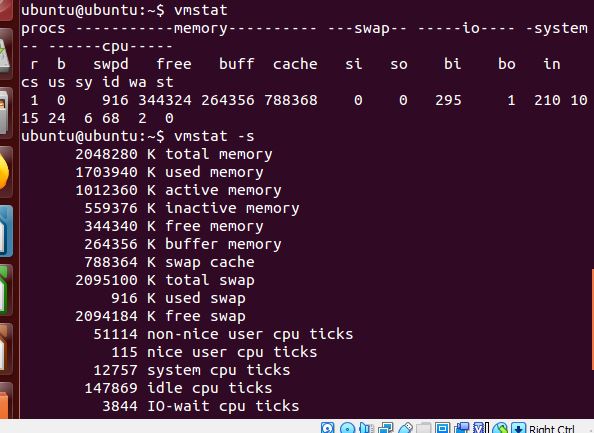


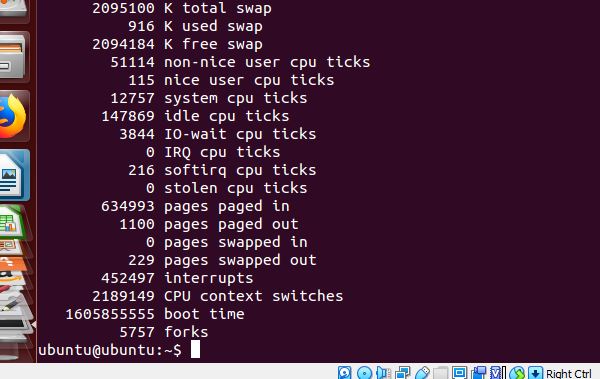
Total : 2048280 Used : 1714612 Free : 333668

1. Baca manual perintah “vmstat”. Option apakah yang harus diberikan jika ingin melihat statistik?

vmstat -s

1. Jalankan perintah vmstat beserta option dari soal sebelumnya. Screenshot hasilnya!





1. Berapa total memori Anda? Berapa yang digunakan? Berapa sisa memori?

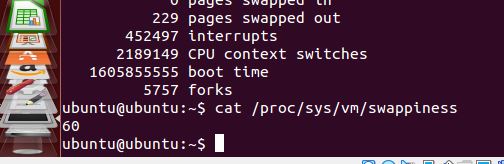
Total : 2048280 Used : 1703940 Free : 344340

1. Apakah hasil antara tool free, meminfo dan vmstat sama atau berbeda? Jika sama berikan buktinya, jika berbeda jelaskan alasan perbedaannya!

Untuk total sama tetapi di used dan free berbeda

1. Swappiness
   1. Apa yang dimaksud dengan swappiness?

* Pertukaran, saat OS melakukan pemindahan proses dari memori ke hardisk
  1. Berapa rentang nilai swappiness?
* Rentangnya adalah antara 0 - 100
  1. Perintah apa yang digunakan untuk mengetahui nilai swappiness pada linux?
* Memakai perintah cat /proc/sys/vm/swappiness
  1. Berapa nilai swappiness komputer Anda? Screenshot hasilnya!
* 60

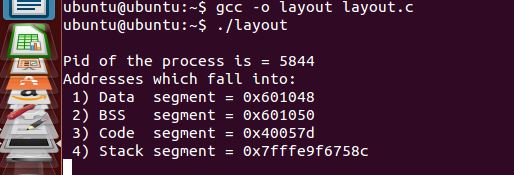


* 1. Perintah apakah yang digunakan untuk mengubah swappiness menjadi 30?
* **Sudo sysctl vm.swappiness=30**

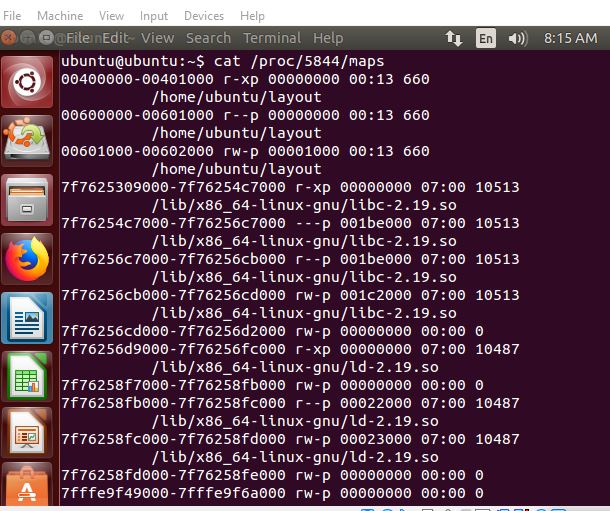
1. Struktur Memori
   1. Lengkapi layout.c berikut ini sehingga menjadi program yang benar!

Done kak 😊

* 1. Compile layout.c dan jalankan program layout! Ambil screenshoot!



* 1. Pada alamat berapa letak kode (text)? (3)0x40057d Pada alamat berapa letak data? (1)0x601048 Pada alamat berapa letak bss? (2)0x601050 Pada alamat berapa letak stack? (4) 0x7fffe9f6758c
  2. Buka terminal baru. Perhatikan PID dari hasil 3.b! Jalankan perintah berikut ini: “cat /proc/PID\_dari\_hasil\_3b/maps” Ambil screenshot!



* 1. Format maps adalah sebagai berikut:

address perms offset dev inode pathname

08048000-08056000 r-xp 00000000 03:0c 64593 /usr/sbin/gpm

Address adalah alamat logika awal dan akhir

Perms adalah ijin rwxp r=read, w=write x=execute p=private s=shared

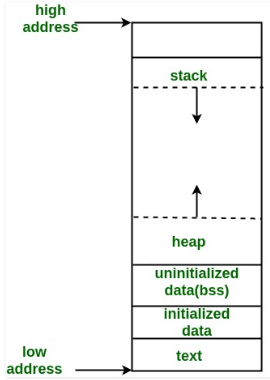
Kode akan mempunyai ijin r dan x saja

Data akan mempunyai ijin r dan w saja

Pada alamat berapa letak kode (text)? 660 Pada alamat berapa letak data? 10513

Pada alamat berapa letak stack? 10487

* 1. Apakah hasil dari layout.c dan maps sama? Jelaskan!
* Untuk data sama tapi berbeda di letak stack
  1. Isikan aalamat sesuai gambar berikut ini:



High address : 7f7625309000 – 7f76254c7000

Stack : 7f76256d9000 – 7f76256fc000

Heap : ox601050

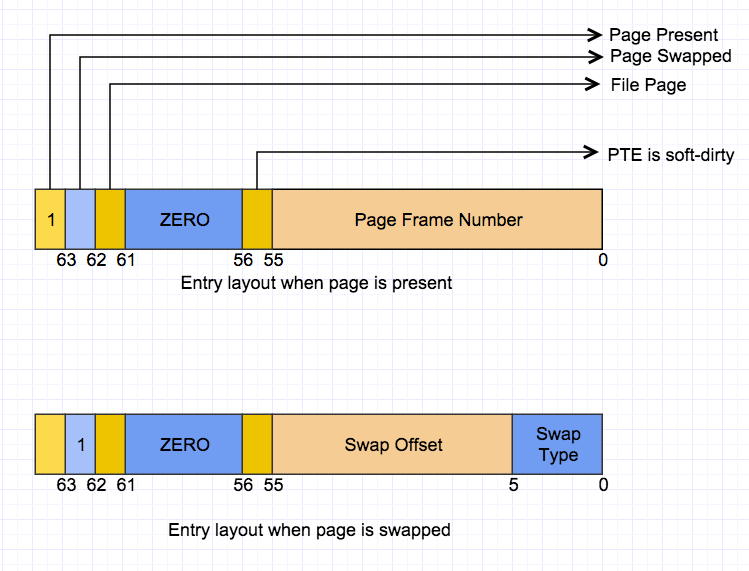
Data : 0x601048

Text : ox40057d

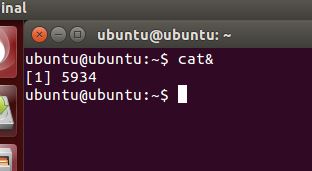
1. Translasi Alamat
   1. Download source code pagemap.c

Page table akan memetakan page dengan frame. Page table dapat diakses pada /proc/pid/pagemap.

Format page table pada pagemap di linux adalah sebagai berikut:

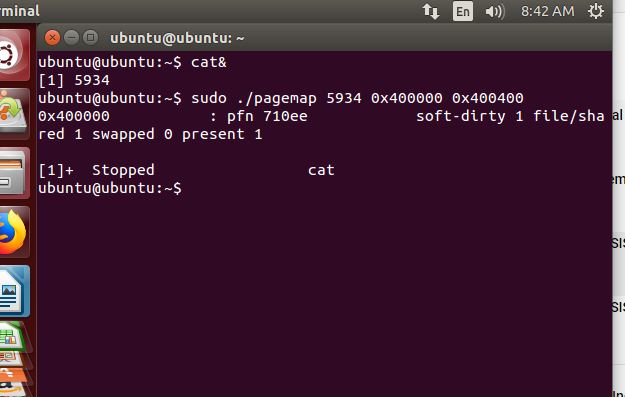


* 1. Compile pagemap.c dan ambil screenshot! Penting: gunakan option -std=c99 **gcc pagemap.c -o pagemap -std=c99**
  2. Buka terminal baru dan jalankan perintah cat&. Ambil screenshot! Catat pidnya!



* 1. Buka terminal dan jalankan pagemap!

“sudo ./pagemap pid\_dari\_nomor\_c 0x400000 0x400400“ Ambil screenshot!



* 1. Berapa page frame number (pfn)? (710ee) Apakah di swap? Iya karena bukan 0
  2. Jelaskan makna dari hasil 4.e!
* Yang dimasukan adalah virtual address 0x400000 - 0x400400 kemudian dari proses shared akan berada pada alamat frame ke 0x400000

Layout.c

//include untuk printf

#include <stdlib.h>

//include untuk menggunakan syscall

#include <unistd.h>

int our\_init\_data; //inisiasi dengan nilai 30

int our\_noinit\_data;

void our\_prints(void)

{

int our\_local\_data = 1;

printf("\nPid of the process is = %d", getpid());

printf("\nAddresses which fall into:");

printf("\n 1) Data segment = %", //type data pointer

&our\_init\_data);

printf("\n 2) BSS segment = %",//type data pointer

&our\_noinit\_data);

printf("\n 3) Code segment = %",//type data pointer

&our\_prints);

printf("\n 4) Stack segment =%\n",//type data pointer

&our\_local\_data);

while(1);

}

int main()

{

// panggil fungsi yang telah dibuat

// return berhasil

}

pagemap.c

#define \_POSIX\_C\_SOURCE 200809L

#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <fcntl.h>

#include <errno.h>

#define PAGE\_SIZE 0x1000

static void print\_page(unsigned long address, unsigned long data) {

printf("0x%-16lx : pfn %-16lx soft-dirty %d file/shared %d "

"swapped %d present %d\n",

address,

data & 0x7fffffffffffff,

(data >> 55) & 1,

(data >> 61) & 1,

(data >> 62) & 1,

(data >> 63) & 1);

}

int main(int argc, char \*argv[]) {

char filename[BUFSIZ];

if(argc != 4) {

printf("Usage: %s pid start\_address end\_address\n",

argv[0]);

return 1;

}

errno = 0;

int pid = (int)strtol(argv[1], NULL, 0);

if(errno) {

perror("strtol");

return 1;

}

snprintf(filename, sizeof filename, "/proc/%d/pagemap", pid);

int fd = open(filename, O\_RDONLY);

if(fd < 0) {

perror("open");

return 1;

}

unsigned long start\_address = strtoul(argv[2], NULL, 0);

unsigned long end\_address = strtoul(argv[3], NULL, 0);

for(unsigned long i = start\_address; i < end\_address; i += 0x1000) {

unsigned long data;

unsigned long index = (i / PAGE\_SIZE) \* sizeof(data);

if(pread(fd, &data, sizeof(data), index) != sizeof(data)) {

perror("pread");

break;

}

print\_page(i, data);

}

close(fd);

return 0;

}