

PRAKTIKUM 7

SISTEM OPERASI

SYSTEM CALL DAN MANAJEMEN MEMORI

NAMA : VERA EFITA HUDI PUTRI

KELAS : SIB 1B

NO ABSEN : 28

PENDAHULUAN

1. Apa yang dimaksud dengan system call?

Jawab : system call adalah cara terprogram di mana program komputer meminta layanan dari kernel sistem operasi tempat dijalankannya. Layanan-layanan ini bisa berupa layanan yang berhubungan dengan hardware (misalnya, mengakses hard disk drive), pembuatan dan pengeksekusian proses baru, dan berkomunikasi dengan layanan kernel integral seperti penjadwalan proses. System calls menyediakan antarmuka antara proses dan sistem operasi.

2. Apa yang dimaksud dengan sistem call fork(), execl() dan wait(). Jawablah dengan menggunakan perintah man (contoh : man 2 fork, man 2 execl dan man 2 wait)?

Jawab :

- man 2 fork, fork() membuat proses baru dengan menduplikasi proses. Proses baru disebut sebagai proses child. Proses yang memanggilnya disebut sebagai proses parent.
- man 2 execl, execl() berfungsi untuk mengeksekusi sebuah file dengan memberi parameter berupa path file dan argumen-argumen perintah file.
- man 2 wait, wait() digunakan untuk menunggu perubahan status pada child process yang dipanggil, dan mendapatkan informasi tentang child process yang statusnya telah berubah.

3. Apa yang dimaksud sistem virtual memory, proses swapping dan buffer cache pada manajemen memory?

Jawab :

- Virtual memory adalah suatu teknik penggunaan disk memori sebagai ekstensi RAM, sehingga ukuran memori RAM bertambah. Kernel akan menulis isi blok memori yang saat ini tidak digunakan ke hard disk sehingga memori dapat digunakan untuk tujuan lain. Ketika memori diperlukan lagi, maka memori tersebut dikembalikan fungsinya.
- Swapping adalah suatu metode pengalihan proses yang bersifat sementara dari

memori utama ke suatu tempat penyimpanan sementara (disk) . Dipanggil Kembali ke memori jika akan melakukan eksekusi.

- Buffer cache adalah memori yang digunakan untuk menyimpan informasi yang hanya bersifat sementara (temporary).

4. Apa yang dimaksud perintah free dan cat /proc/meminfo?

Jawab :

- Perintah free adalah perintah yang dapat menampilkan informasi tentang memori RAM dan swap dalam kibibyte (1 kilobyte (KB) = 1024 bytes)
- cat /proc/meminfo memiliki fungsi yang sama dengan perintah free, hanya saja informasi yang ditampilkan lebih lengkap.

5. Apa yang dimaksud perintah ps?

Jawab :

Perintah ps digunakan untuk menampilkan informasi tentang proses yang berjalan di sistem, seperti ID proses, pengguna yang menjalankan proses, penggunaan CPU, dan penggunaan memori. Dengan menggunakan argumen tertentu, ps dapat menampilkan informasi yang lebih terperinci tentang proses yang berjalan.

PERCOBAAN 7 : Melihat Manajemen Memori

1. Perhatikan dengan perintah dmesg jumlah memory tersedia dan proses swapping

\$ dmesg | more

```
verefita@verefita-virtual-machine:~$ dmesg | more
dmesg: read kernel buffer failed: Operation not permitted
verefita@verefita-virtual-machine:~$
```

2. Dengan perintah free perhatikan jumlah memory "free", "used", "share" dan "buffer".

\$ free

```
verefita@verefita-virtual-machine:~$ free
              total        used        free      shared  buff/cache   available
Mem:            1968552         965092         184232         25040         819228         813248
Swap:            2191356         18264         2173072
verefita@verefita-virtual-machine:~$
```

3. Dengan perintah dibawah ini apakah hasilnya sama dengan no 2 ?tidak

\$ cat /proc/meminfo

```
verefita@verefita-virtual-machine:~$ cat /proc/meminfo
MemTotal:       1968552 kB
MemFree:        162144 kB
MemAvailable:   811152 kB
Buffers:        18316 kB
Cached:         744176 kB
SwapCached:     8200 kB
Active:         276380 kB
Inactive:       1088700 kB
Active(anon):   172168 kB
Inactive(anon): 443612 kB
Active(file):   104212 kB
Inactive(file): 645688 kB
Unevictable:    0 kB
Mlocked:        0 kB
SwapTotal:      2191356 kB
SwapFree:       2165088 kB
Zswap:          0 kB
Zswapped:        0 kB
Dirty:          388 kB
Writeback:       0 kB
AnonPages:      594360 kB
Mapped:         300404 kB
Shmem:          12668 kB
KReclaimable:   60636 kB
```

4. Gunakan perintah dibawah ini

\$ ls -lR /.

```
drwxr-xr-x 2 root root 59 Mar 30 2023 libthai-data
drwxr-xr-x 2 root root 59 Mar 30 2023 libtiff5
drwxr-xr-x 2 root root 59 Mar 30 2023 libunwind8
drwxr-xr-x 2 root root 59 Mar 30 2023 libwayland-client0
drwxr-xr-x 2 root root 59 Mar 30 2023 libwayland-cursor0
drwxr-xr-x 2 root root 59 Mar 30 2023 libwayland-egl1
drwxr-xr-x 2 root root 59 Mar 30 2023 libwebp7
drwxr-xr-x 2 root root 96 Mar 30 2023 libx11-6
drwxr-xr-x 2 root root 59 Mar 30 2023 libx11-data
drwxr-xr-x 2 root root 59 Mar 30 2023 libxau6
drwxr-xr-x 2 root root 59 Mar 30 2023 libxcb1
drwxr-xr-x 2 root root 59 Mar 30 2023 libxcb-render0
drwxr-xr-x 2 root root 59 Mar 30 2023 libxcb-shm0
drwxr-xr-x 2 root root 59 Mar 30 2023 libxcomposite1
drwxr-xr-x 2 root root 59 Mar 30 2023 libxcursor1
drwxr-xr-x 2 root root 59 Mar 30 2023 libxdamage1
drwxr-xr-x 2 root root 59 Mar 30 2023 libxdmcp6
drwxr-xr-x 2 root root 59 Mar 30 2023 libxext6
drwxr-xr-x 2 root root 59 Mar 30 2023 libxfixed3
drwxr-xr-x 2 root root 59 Mar 30 2023 libxi6
drwxr-xr-x 2 root root 59 Mar 30 2023 libxinerama1
drwxr-xr-x 2 root root 59 Mar 30 2023 libxkbcommon0
drwxr-xr-x 2 root root 112 Mar 30 2023 libxml2
drwxr-xr-x 2 root root 59 Mar 30 2023 libxmlb2
drwxr-xr-x 2 root root 59 Mar 30 2023 libxrandr2
```

5. Perhatikan perubahan manajemen memory

\$ free

```
verefita@verefita-virtual-machine:~$ free
              total        used        free      shared  buff/cache   available
Mem:           1968552       888428       185824        13392        894300        879124
Swap:          2191356       124772       2066584

verefita@verefita-virtual-machine:~$
```

6. Jalankan sebuah program, misalnya open Office. Perhatikan perubahan manajemen memory

\$ free

```
verefita 4062 0.0 0.1 12940 3584 pts/1 R+ 10:55 0:00 ps -uax
verefita@verefita-virtual-machine:~$ free
              total        used        free      shared  buff/cache   available
Mem:           1968552        875792        282428         15964         810332        889272
Swap:          2191356        133420         2057936
```

7. Dengan perintah ps bagaimana penggunaan memory untuk se tiap proses diatas ?

\$ ps -uax

```
verefita@verefita-virtual-machine:~$ ps -uax
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.6  0.6 102580 13296 ?        Ss   10:34   0:08 /sbin/i
root         2  0.0  0.0      0      0 ?        S    10:34   0:00 [kthrea
root         3  0.0  0.0      0      0 ?        I<   10:34   0:00 [rcu_gp
root         4  0.0  0.0      0      0 ?        I<   10:34   0:00 [rcu_pa
root         5  0.0  0.0      0      0 ?        I<   10:34   0:00 [slub_f
root         6  0.0  0.0      0      0 ?        I<   10:34   0:00 [netns]
root         8  0.0  0.0      0      0 ?        I<   10:34   0:00 [kworke
root        11  0.0  0.0      0      0 ?        I<   10:34   0:00 [mm_per
root        12  0.0  0.0      0      0 ?        I    10:34   0:00 [rcu_ta
root        13  0.0  0.0      0      0 ?        I    10:34   0:00 [rcu_ta
root        14  0.0  0.0      0      0 ?        I    10:34   0:00 [rcu_ta
root        15  0.0  0.0      0      0 ?        S    10:34   0:00 [ksofti
root        16  0.1  0.0      0      0 ?        I    10:34   0:01 [rcu_pr
root        17  0.0  0.0      0      0 ?        S    10:34   0:00 [migrat
root        18  0.0  0.0      0      0 ?        S    10:34   0:00 [idle_i
root        19  0.0  0.0      0      0 ?        S    10:34   0:00 [cpuhp/
root        20  0.0  0.0      0      0 ?        S    10:34   0:00 [cpuhp/
root        21  0.0  0.0      0      0 ?        S    10:34   0:00 [idle_i
root        22  0.0  0.0      0      0 ?        S    10:34   0:00 [migrat
root        23  0.0  0.0      0      0 ?        S    10:34   0:00 [ksofti
root        26  0.0  0.0      0      0 ?        S    10:34   0:00 [kdevtm
root        27  0.0  0.0      0      0 ?        I<   10:34   0:00 [inet_f
root        29  0.0  0.0      0      0 ?        S    10:34   0:00 [kaudit
root        31  0.0  0.0      0      0 ?        S    10:34   0:00 [khungt
root        33  0.0  0.0      0      0 ?        S    10:34   0:00 [oom_re
Trash      34  0.0  0.0      0      0 ?        I<   10:34   0:00 [writeb
root        35  0.4  0.0      0      0 ?        S    10:34   0:05 [kcompa
root        36  0.0  0.0      0      0 ?        SN   10:34   0:00 [ksmd]
root        39  0.0  0.0      0      0 ?        SN   10:34   0:00 [khugep
root        40  0.0  0.0      0      0 ?        I<   10:34   0:00 [kinteg
root        41  0.0  0.0      0      0 ?        I<   10:34   0:00 [kblock
root        42  0.0  0.0      0      0 ?        I<   10:34   0:00 [blkcg_
root        43  0.0  0.0      0      0 ?        I<   10:34   0:00 [tpm_de
root        44  0.0  0.0      0      0 ?        I<   10:34   0:00 [ata_sf
```

LATIHAN

6. Lakukan hal yang sama dengan percobaan 7 untuk melihat perubahan memory setelah dilakukan beberapa proses pada shell. Tentukan perintah yang dilakukan misalnya membuka browser dan perhatikan hal-hal berikut :

- a. Informasi apa saja yang ditampilkan dengan perintah free ?

jawab : Perintah free menampilkan informasi tentang total memori, memori yang sedang digunakan, dan memori yang tersedia pada sistem.

```
erefit@verefita-virtual-machine:~$ free
              total        used        free      shared  buff/cache   available
mem:          1968588         996056        153788         40128         818744        769928
swap:          2191356           0        2191356
erefit@verefita-virtual-machine:~$ S
```

- b. Informasi apa saja yang disimpan file /proc/meminfo ?

Jawab :

- MemTotal: Total memori fisik pada sistem.
- MemFree: Jumlah memori fisik yang tersedia saat ini.
- MemAvailable: Jumlah memori fisik yang tersedia untuk alokasi tambahan.
- Buffers: Memori yang digunakan untuk buffer data.
- Cached: Memori yang digunakan untuk menyimpan cache data.
- SwapTotal: Total ruang swap yang tersedia.
- SwapFree: Jumlah ruang swap yang tersedia saat ini.
- AnonPages: Jumlah memori anonim yang dialokasikan.
- Mapped: Jumlah memori yang dipetakan ke file.
- Shmem: Jumlah memori bersama yang digunakan.
- Slab: Memori yang digunakan untuk objek kernel.
- CommitLimit: Batas total komitmen memori yang diperbolehkan.
- Committed_AS: Jumlah total memori yang dialokasikan oleh sistem.
- Dan informasi lainnya seperti KernelStack, PageTables, Percpu, dan lainnya yang berkaitan dengan alokasi dan penggunaan memori sistem.

- c. Berapa besar kapasitas memory total ?

Jawab :

Kapasitas memori total dapat ditemukan dalam output dari perintah cat /proc/meminfo, biasanya di bawah entri "MemTotal".


```
verefita@verefita-virtual-machine:~$ cat /proc/meminfo | grep MemTotal
MemTotal:      1968588 kB
verefita@verefita-virtual-machine:~$
```

- d. Berapa kapasitas memory yang sudah terpakai ?

Jawab : Kapasitas memori yang sudah terpakai dapat dilihat dalam output dari perintah free, dalam kolom "used".

1872656 KB

- e. Berapa kapasitas memory yang belum terpakai ?

Jawab : free dalam kolom free

```
verefita@verefita-virtual-machine:~$ cat /proc/meminfo | grep MemFree
MemFree:      95932 kB
verefita@verefita-virtual-machine:~$
```

- f. Berapa kapasitas memory yang digunakan sharing beberapa proses?

Jawab :

```
verefita@verefita-virtual-machine:~$ cat /proc/meminfo | grep Shmem
Shmem:      33576 kB
ShmemHugePages:    0 kB
ShmemPmdMapped:    0 kB
verefita@verefita-virtual-machine:~$
```

- g. Berapa kapasitas buffer cache ?

Jawab :

```
verefita@verefita-virtual-machine:~$ cat /proc/meminfo | grep -iE "buffer|
cache"
Buffers:      40124 kB
verefita@verefita-virtual-machine:~$
```

KESIMPULAN

System calls adalah cara terprogram di mana program komputer meminta layanan dari kernel sistem operasi tempat dijalankannya. System calls menyediakan antarmuka antara proses dan sistem operasi. Pada praktikum kali ini, kita melakukan 3 macam system calls, yaitu system calls fork, execl, dan wait. System calls fork adalah system calls yang dapat membuat proses baru dengan menduplikasi proses. Proses baru disebut sebagai proses child. Proses yang memanggilnya disebut sebagai proses parent. System calls execl berfungsi untuk mengeksekusi sebuah file dengan memberi parameter berupa path file dan argumen-argumen perintah file. Sedangkan system calls wait adalah ystem call yang digunakan untuk menunggu perubahan status pada child process yang dipanggil, dan mendapatkan informasi tentang child process yang statusnya telah berubah.