**Nama : Deanissa Sherly Sabilla**

**Kelas / Absen : SIB 1B / 06**

**PERCOBAAN:**

1. Login sebagai user.
2. Jalankan Kalkulator

* Menggunakan perintah ‘bc’ untuk menjalankan kalkulator



1. Lakukan percobaan-percobaan di bawah ini kemudian analisa hasil percobaan.
2. Selesaikan soal-soal latihan.

# Percobaan 1 : Status Proses

|  |
| --- |
| 1. Instruksi ps (*process status*) digunakan untuk melihat kondisi proses yang ada. PID adalah Nomor Identitas Proses, TTY adalah nama terminal dimana proses tersebut aktif, STAT berisi S (*Sleepin* g) dan R (*Running*), COMMAND merupakan instruksi yang digunakan.  * $ *ps*      1. Untuk melihat fak tor/elemen lainnya, gunakan option –u (user). %CPU adalah presentasi CPU time yang digunakan oleh proses tersebut, %MEM adalah presentasi system memori yang digunakan proses, SIZE adalah jumlah memori yang digunakan, RSS (*Real System Storage*) adalah jumlah memori yang digunakan, START adalah kapan proses tersebut diaktifkan   $ *ps -u <user>*   1. Mencari proses yang spesifik pemakai. Proses diatas hanya terbatas pada proses milik pemakai, dimana pemakai teresbut melakukan login  * $ *ps –u <user>*      1. Mencari proses lainnya gunakan opsi a (*all*) dan au (*all user*)  * $ *ps –a*      * $ *ps –au* |

**Percobaan 2 : Menampilkan Hubungan Proses Parent dan Child**

|  |
| --- |
| 1. Ketik **ps –eH** dan tekan **Enter**. Opsi **e** memilih semua proses dan opsi **H** menghasilkan tampilan proses secara hierarki. Proses child muncul dibawah proses parent. Proses child ditandai dengan awalan beberapa spasi.  * $ *ps -eH*      1. Ketik **ps –e f** dan tekan **Enter**. Tampilan serupa dengan langkah 2. Opsi **–f** akan menampilkan status proses dengan karakter grafis (\ dan \_)  * $ *ps –e f*      1. Ketik **pstree** dan tekan **Enter**. Akan ditampilkan semua proses pada sistem dalam bentuk hirarki parent/child. Proses parent di sebelah kiri proses child. Sebagai contoh proses init sebagai parent (*ancestor*) dari semua proses pada sistem. Beberapa child dari init mempunyai child. Proses login mempunya i proses bash sebagai child. Proses bash mempunyai proses child startx. Proses startx mempunyai child xinit dan seterusnya. $ *pstree*    1. Ketik **pstree | grep mingetty** dan tekan **Enter**. Akan menampilkan semua proses mingetty yang berjalan pada system yang berupa *console virtual*. Selain menampikan semua proses, proses dikelompokkan dalam satu baris dengan suatu angka sebagai jumlah proses yang berjalan.  * $ *pstree | grep <kalkulator>*      1. Untuk melihat semua PID untuk proses gunakan opsi **–p**.  * $ *pstree –p*      1. Untuk menampilk an proses dan ancestor yang tercetak tebal gunakan opsi **–h**.  * $ *pstree –h* |

**Percobaan 3 : Menampilkan Status Proses dengan Berbagai Format**

|  |
| --- |
| 1. Ketik **ps –e | more** dan tekan **Enter**. Opsi -**e** menampilkan semua proses dalam bentuk 4 kolom : PID, TTY, TIME dan CMD.  * $ *ps –e | more*      1. Ketik **ps ax | more** dan tekan **Enter**. Opsi **a** akan menampilkan semua proses yang dihasilkan terminal (TTY). Opsi **x** menampilkan semua proses yang tidak dihasilkan terminal. Secara logika opsi ini sama dengan opsi **–e** . Terdapa 5 kolom : PID, TTY, STAT, TIME dan COMMAND.  * $ *ps ax | more*      1. Ketik **ps –e f | more** dan tekan **Enter**. Opsi **–e f** akan menampilkan semua proses dalam format daftar penuh.  * $ *ps ef | more*      1. Ketik **ps –eo pid, cmd | more** dan tekan **Enter**. Opsi **–eo** akan menampilkan semua proses dalam format sesuai definisi user yaitu terdiri dari kolom PID dan CMD.  * $ *ps –eo pid,cmd | more* |

|  |
| --- |
| 20. Ketik **ps –eo pid,ppid,%mem,cmd | more** dan tekan **Enter**. Akan menampilkan kolom PID, PPID dan %MEM. PPID adalah proses ID dari proses parent. %MEM menampilkan persentasi memory system yang digunakan proses. Jika proses hanya menggunakan sedikit memory system akan dita mpilkan 0.   * $ *ps –eo pid,ppid,%mem,cmd | more* |

# Percobaan 4 : Mengontrol proses pada shell

|  |
| --- |
| 1. Gunakan perintah yes yang mengirim output y yang tidak pernah berhenti  * $ *yes*      1. Belokkan standart output ke /dev/null  * $ *yes > /dev/null*      1. Salah satu cara agar perintah yes tetap dijalankan tetapi shell tetap digunakan untuk hal yang lain dengan meletakkan proses pada *background* dengan menambahkan karakter & pada akhir perintah.  * $ *yes > /dev/null &*   Angka dalam ”[ ]” merupakan **job number** diikuti PID.   1. Untuk melihat status proses gunakan perintah jobs.  * $ *jobs*     7. Untuk menghentikan job, gunakan perintah kill diikuti *job number* atau PID proses. Untuk identifikasi job number, diikuti prefix dengan karakter ”%”. Pilih nomor job untuk genome kalkulator   * $ *kill %<nomor job>* contoh : kill %1   4. Lihat status job setelah diterminasi   * $ *jobs* |