

Logbook Minggu 11

Materi yang sudah dipahami :

- *Daemon process*

Daemon process adalah proses yang umumnya berjalan dari awal komputer hidup hingga komputer mati. *Daemon process* berjalan di *background* dan tidak memiliki *controlling terminal*. Contoh *daemon process* antara lain adalah cron, sshd, httpd, dan inetd.

- Cara menjalankan proses di *foreground* dan *background*

Untuk menjalankan proses di *background*, tambahkan & di akhir *command* (misal : `./loop.sh &`). Untuk melihat *job* yang sedang dijalankan oleh *shell* (termasuk *background process*), dapat menggunakan *command jobs*. Untuk mengubah *background process* menjadi *foreground process*, jalankan *command fg <job id>*. Secara umum, *background process* akan ikut *terminate* saat *parent*-nya *terminate*, kecuali jika dibuat *immune* (dapat dicapai dengan menggunakan *nohup*, *disown*, *tmux*, *screen*).

- Perbandingan antara *nohup* dan *disown*

- *Command nohup* tidak menghapus proses dari *job control* milik *shell*, sementara *disown* akan menghapus proses dari *job control* milik *shell*.
- Dengan *command nohup*, *standard output* dan *standard error* dari proses ini akan di-*redirect* ke *nohup.out* (*by default*), dapat juga di-*redirect* ke file lain, contohnya dengan menggunakan *command nohup ./loop.sh > output_file.out &*. Pada *disown*, *standard output* dan *standard error* akan diarahkan ke terminal selama *shell* belum *terminate*.
- Keduanya membuat proses yang dijalankan menjadi *immune* terhadap *hangups*.

- *Process group*, *session*, dan hubungan antar keduanya

Process group adalah kumpulan dari satu atau beberapa proses. Suatu *process group* diidentifikasi berdasarkan *process group ID* miliknya. *Process group ID* umumnya akan sama dengan *process ID* dari anggota pertama dari *process group* tersebut. *Process group* sering digunakan untuk mengatur distribusi sinyal. Apabila suatu sinyal diarahkan ke *process group* tertentu, setiap proses dalam *process group* tersebut akan menerima sinyal tersebut. Pada *shell* yang menerapkan *job control*, setiap *command* pada terminal umumnya akan membentuk *process group* baru. Apabila dalam eksekusi *command* tersebut terbentuk proses-proses baru, maka proses tersebut akan masuk ke dalam *process group* yang sama.

Session merupakan kumpulan dari satu atau beberapa *process group* yang terikat ke satu *controlling terminal* yang sama (atau tidak terikat ke terminal manapun, contohnya *daemon*). Semua *process group* yang dibuat di *shell* yang sama akan memiliki *session ID* yang sama. Dalam satu *session*, terdapat satu *process group* yang berjalan di *foreground*. Setiap sinyal yang di-*generate* di terminal (CTRL-C, CTRL-Z, dll.) akan diarahkan ke *foreground process group* ini. Selain itu, hanya proses-proses yang termasuk ke *foreground process group* yang dapat membaca input dari terminal. Suatu proses hanya dapat berpindah ke *process group* lain yang terletak pada *session* yang sama.

Video yang sudah ditonton :

- Module 10-01/01-Daemon & Background Process