

Logbook Minggu 4

Materi yang sudah dipahami :

- Perbedaan program dan proses

Program merupakan sekumpulan instruksi untuk menyelesaikan *task* tertentu, sementara *process* merupakan entitas yang dibentuk ketika suatu program dieksekusi.

- Definisi *environment variable*

Environment variable adalah variabel yang didefinisikan di awal sebelum program dijalankan.

- Alasan *counter PID* di-reset kembali ke 300 saat sistem sudah mencapai *PID* tertinggi *PID* 0-299 sudah di-reserve untuk proses-proses khusus kernel atau OS.

- Perbedaan *zombie*, *orphaned*, dan *daemon process*

Zombie process adalah proses yang sudah selesai dieksekusi tetapi masih ada di *process table*, umumnya terjadi pada *child process* yang masih menunggu *parent*-nya membaca exit status melalui *wait system call*.

Orphaned process adalah proses yang masih tetap berjalan di saat *parent process* nya sudah berhenti, *orphaned process* ini kemudian akan diadopsi oleh init process.

Daemon process adalah proses yang berjalan di *background* dan terbebas dari kontrol langsung oleh user, *daemon process* merupakan salah satu jenis *orphaned process* yang disengaja.

- Cara mencegah terbentuknya *zombie process*

Mengeksekusi *wait()* system call pada *parent process*, mengabaikan sinyal SIGCHLD, memasang *handler* untuk sinyal SIGCHLD pada *parent process*, menerapkan *double forking*.

- Mekanisme *copy-on-write* pada *fork()*

Ketika *fork()* dieksekusi, akan dibentuk *child process* baru dengan *parent*-nya adalah proses yang melakukan pemanggilan *fork()*. Kedua proses ini pada awalnya akan berbagi halaman-halaman yang sama di memori, dengan halaman-halaman tersebut ditandai sebagai *copy-on-write*. Apabila salah satu proses kemudian hendak melakukan modifikasi pada halaman tertentu, maka halaman tersebut akan diduplikasi dan modifikasi akan dilakukan pada halaman hasil duplikasi. Selanjutnya, proses yang melakukan modifikasi akan *me-refer* ke halaman hasil duplikasi, sementara proses yang satunya tetap *me-refer* ke halaman awal. Adapun halaman-halaman lain tetap *shared* di antara kedua proses.

- Perbedaan *exit()* dan *_exit()*

exit() akan terlebih dahulu memanggil fungsi yang diberikan di fungsi *atexit()* dan melakukan *cleanup* (termasuk di antaranya menutup semua *stream* dan menghapus semua *temporary file*) sebelum benar-benar memberhentikan proses, sedangkan *_exit()* tidak melakukan dua tahapan tersebut.

Video yang sudah ditonton :

- Module 04-01/02-Process&Memory
- Module 04-02/02-Process&Memory