

Week 3 Logbook

Aktivitas Minggu ini

Pada minggu ini, saya melakukan hal-hal di bawah ini untuk meningkatkan pengetahuan dan pengalaman saya dalam System Programming:

1. Mengerjakan Pre Test
2. Mengerjakan WS3
3. Menonton video materi di Youtube
4. Menghadiri kelas Sysprog pada hari Jum'at tanggal 2 Oktober 2020
5. Mencari sumber referensi tentang stat system call, file descriptor, daemon, dan lain-lain
6. Mencoba-coba command serta program pada Linux
7. Berdiskusi serta menyamakan pengetahuan dengan teman mengenai materi minggu ini
8. Membaca slide yang sudah disiapkan di Scele

File Descriptor

Sebuah entry berbentuk integer non negative yang disimpan pada kernel berfungsi untuk merepresentasikan file dan informasi tentang file yang sedang dibuka.

Inode Table

Sebuah data struktur yang menyimpan semua informasi tentang suatu file seperti tipe file, permission akses, atribut, time stamp dan lain-lain.

Stat Structure

stat structure adalah sebuah struktur dari suatu sistem operasi yang didefinisikan untuk menyimpan informasi umum tentang suatu file. Informasi yang disimpan oleh struktur ini antara lain:

1. ``st_mode`` - permission dari file
2. ``st_inode`` - inode untuk file
3. ``st_dev`` - di device mana file tersebut berada
4. ``st_uid`` - user id file
5. ``st_gid`` - group id file
6. ``st_atime`` - waktu terakhir file diakses
7. ``st_ctime`` - waktu terakhir permission dari file diganti

8. ``st_mtime`` - waktu terakhir isi file diubah
9. ``st_nlink`` - jumlah hard link yang pada file
10. ``st_size`` - size dari file

Daemon

Daemon adalah suatu program pada sistem operasi yang dijalankan di background, ketimbang dikendalikan langsung oleh pengguna. Daemon juga tidak memiliki parent process karena daemon hanya mewariskan pid proses `init`, namun `init` bukanlah parent dari daemon. Daemon juga tidak dapat dikontrol dari terminal di komputer kita sehingga biasanya daemon tidak memiliki suatu file descriptor.

Mekanisme copy pada Linux

Pertama program akan meminta 2 input, yaitu file program yang akan di copy dan file baru yang merupakan hasil copy dari file sebelumnya. Lalu kita menjalankan `open` file program yang akan di copy agar mendapatkan file descriptornya dan agar kita hanya bisa membaca filenya saja. Setelah itu, program akan membuat sebuah variabel yang isinya merupakan file descriptor hasil eksekusi `creat` atau `open` dengan flag tertentu yang mana variabel ini merupakan bakal jadi file hasil copy dari file sebelumnya. Setelah itu, program akan me-

loop sampai byte yang dibaca oleh system call `read` sama dengan nol. Byte ini nantinya akan dikurangi terus berdasarkan jumlah buffer yang diinginkan. Pengurangan byte ini dilakukan oleh system call `read` karena system call ini akan terus menghitung byte dari file descriptor untuk disimpan ke dalam buffer sehingga panjang byte yang dibaca makin kecil sampai menuju akhir file, yaitu offsetnya 0. Di dalam loop while tersebut, file baru akan ditulis isinya menggunakan system call `write` dengan parameter file descriptor dari file output baru, buffer saat ini yang telah disimpan oleh `read`, dan panjang byte file loop saat ini. Setelah selesai, program akan exit sukses.

Universality of I/O

Pada sistem operasi, kita dapat melakukan copy-paste berbagai hal mulai dari file, device output, sampai terminal. Ternyata copy itu merupakan sebuah program yang diletakkan pada root folder dari unix/linux sehingga kita dapat dengan bebas melakukan copy di mana saja, kapan saja.

Perbedaan creat() dan open()

Perbedaan antara creat dan open ialah creat lebih berfungsi untuk membuat suatu file baru ketimbang open yang bisa dengan bebas membuat file baru atau membukanya saja. Jika creat dilakukan pada file yang sudah ada sebelumnya, file tersebut akan ditulis ulang dari awal sehingga isinya pun menghilang.