

Rocky Arkan Adnan Ahmad

1806186566

SysProg-A

Hal yang sudah dipelajari dan dipahami:

1. I/O Buffering

Buffer dalam I/O adalah sebuah memory yang dimana sebuah data akan disimpan untuk sementara waktu sampai data tersebut akan baru diwrite ke dalam hardware.

Buffer size akan mempengaruhi kecepatan read dan write(). Semakin besar buffer size, maka semakin cepat operasi I/O. Namun, buffer size juga menggunakan memori, sehingga semakin besar buffer size maka semakin besar juga memori yang digunakan. Jika buffer size terlalu kecil maka kecepatan operasi I/O juga menjadi sangat lambat, sehingga kita harusnya mencari buffer size optimal dimana mendapatkan kecepatan yang cukup cepat dan juga menggunakan memori yang tidak terlalu banyak.

2. Mekanisme I/O Buffering

Pertama-tama, user data yang diinginkan masuk ke dalam fungsi library stdio seperti printf(). Lalu, fungsi library stdio tersebut masuk ke dalam buffernya stdio, yang diatur dalam user memory space. Ketika buffer dipenuhi, maka fungsi library akan memanggil I/O system call seperti write(), sehingga user data yang ada di fungsi library pindah ke system call. System call tersebut juga lalu memasukan data ke dalam buffer cache yang diatur oleh kernel. Setelah itu, kernel lalu menulis datanya ke dalam disk atau hardware.

Pemanggilan system call atau penulisan ke dalam hardware dapat dilakukan kapan saja untuk mengef flush buffer stdio, buffer kernel, atau keduanya. Lalu, setiap pemanggilan I/O dapat mengef flush buffer secara automatic. Bisa dengan memanggil setbuff(stream, NULL) sehingga buffer di stdio didisable, atau dengan open() dengan flag O_SYNC untuk membuat system call menjadi synchronous, sehingga setiap write() akan langsung diflush ke dalam hardware.

3. Makefile

Makefile adalah sebuah file yang berisi rule-rule yang dapat digunakan untuk mengautomasi sesuatu. Makefile dapat digunakan dengan menggunakan command make, dimana dia akan mengeksekusi target default (yaitu target paling awal). Kalau command dari target kosong, maka dia akan secara default mengeksekusi cc -o <nama_target>.c <nama_target>, atau dengan kata lain mengcompile program c dengan nama yang sama dengan nama target.

4. Mekanisme write() dan read()

Ketika mengeksekusi read(), maka akan menyimpan karakter yang dibaca di argument pertama read ke dalam sebuah array char atau buffer yang diberi di argument ke dua, sebanyak jumlah yang didefine di argument ketiga. Lalu read() akan memindahkan seek sampai index dia membaca karakter terakhir dari file. Lalu, write akan membaca array / buffer yang diberi di argumen ke dua, lalu menulisnya sebanyak yang didefine di argument ke tiga, dan menulisnya

ke file descriptor yang didefine di argument pertama. Ketika memanggil write(), maka tidak langsung ditulis ke hardware, tapi masuk ke dalam kernel buffer cache terlebih dahulu.

Hal yang dipelajari namun belum dipahami:

Untuk minggu ini belum ada materi yang belum saya pahami.