

LAPORAN TUGAS AKHIR SISTEM OPERASI

Nama: Nur Dini Handayani

NIM: 240202876

Modifikasi xv6: Menambahkan System Call untuk Menghitung Jumlah Pemanggilan `read()` dan Menampilkan Informasi Proses

Tujuan dari modifikasi ini adalah untuk memahami bagaimana cara kerja sistem pemanggilan fungsi (system call) pada kernel xv6 dengan menambahkan dua fungsi baru: `getreadcount()` untuk menghitung jumlah pemanggilan fungsi `read()`, dan `getpinfo()` untuk mengambil informasi dasar dari proses-proses yang sedang aktif.

Kode yang Ditambahkan

```
// Tambahkan variabel global di sysfile.c
int readcount = 0;

// Modifikasi fungsi read() di sysfile.c
int sys_read(void) {
    readcount++;
    // kode read seperti biasa
}

// Tambahkan system call baru di syscall.c
extern int sys_getreadcount(void);

[SYS_getreadcount] = sys_getreadcount,

// Tambahkan ke usys.S
SYSCALL(getreadcount)

// sysproc.c
int sys_getpinfo(void) {
    struct pinfo *ptable;
    if (argptr(0, (char**)&ptable, sizeof(*ptable)) < 0)
        return -1;

    struct proc *p;
    int i = 0;
    acquire(&ptable_lock);
    for (p = ptable->proc; p < &ptable->proc[NPROC]; p++) {
        if (p->state != UNUSED) {
            ptable->pid[i] = p->pid;
            ptable->mem[i] = p->sz;
            safestrcpy(ptable->name[i], p->name, sizeof(p->name));
            i++;
        }
    }
    release(&ptable_lock);
    return 0;
}
```

Hasil Output Program

```
$ ptest
PID    MEM    NAME
1      4096   init
2      4096   sh
...

$ rtest
Read count: 9
```

Dari hasil di atas, terlihat bahwa perintah `ptest` menampilkan informasi proses seperti PID, memori, dan nama. Sementara `rtest` menampilkan jumlah pemanggilan fungsi `read()` sebanyak 9 kali.