**📝 Laporan Tugas Akhir**

**Mata Kuliah**: Sistem Operasi  
**Semester**: Genap / Tahun Ajaran 2024–2025  
**Nama**: Panji Kurniawan  
**NIM**: 240202904  
**Modul yang Dikerjakan**:  
Modul 1 – System Call dan Instrumentasi Kernel

**📌 Deskripsi Singkat Tugas**

**Modul 1 – System Call dan Instrumentasi Kernel**  
Saya menambahkan dua system call baru ke dalam kernel xv6:

* getpinfo(): mengambil informasi proses-proses aktif dari kernel ke user space, mencakup PID dan status proses.
* getReadCount(): menghitung berapa kali fungsi read() dipanggil sejak sistem dinyalakan.

Tujuan modul ini adalah memahami bagaimana membuat system call baru dan melakukan instrumentasi sederhana terhadap kernel.

**🛠️ Rincian Implementasi**

Langkah-langkah yang saya lakukan:

1. **Menambahkan system call getpinfo()**
   * Membuat struktur baru struct pinfo di proc.h untuk menyimpan informasi proses.
   * Menambahkan logika pengisian informasi proses dari ptable[] dalam fungsi sys\_getpinfo() di sysproc.c.
   * Memastikan pointer yang dikirim dari user space divalidasi dengan argptr().
2. **Menambahkan system call getReadCount()**
   * Menambahkan variabel global int readcount di sysfile.c.
   * Menambahkan readcount++ pada setiap pemanggilan sys\_read().
   * Implementasi syscall sys\_getreadcount() untuk mengembalikan nilai readcount.
3. **Mendaftarkan syscall**
   * syscall.h, user.h, usys.S, dan syscall.c ditambahkan entri baru untuk kedua syscall.
4. **Program uji**
   * ptest.c: program user untuk mencetak informasi semua proses aktif menggunakan getpinfo().
   * rtest.c: mencetak jumlah pemanggilan read() menggunakan getReadCount().

**✅ Uji Fungsionalitas**

Program yang digunakan:

* ptest: menunjukkan daftar proses beserta statusnya.
* rtest: menunjukkan berapa kali read() dipanggil sejak boot.

**📷 Hasil Uji**

**📍 Output ptest:**

$ ptest

PID MEM NAME

1 12288 init

2 16384 sh

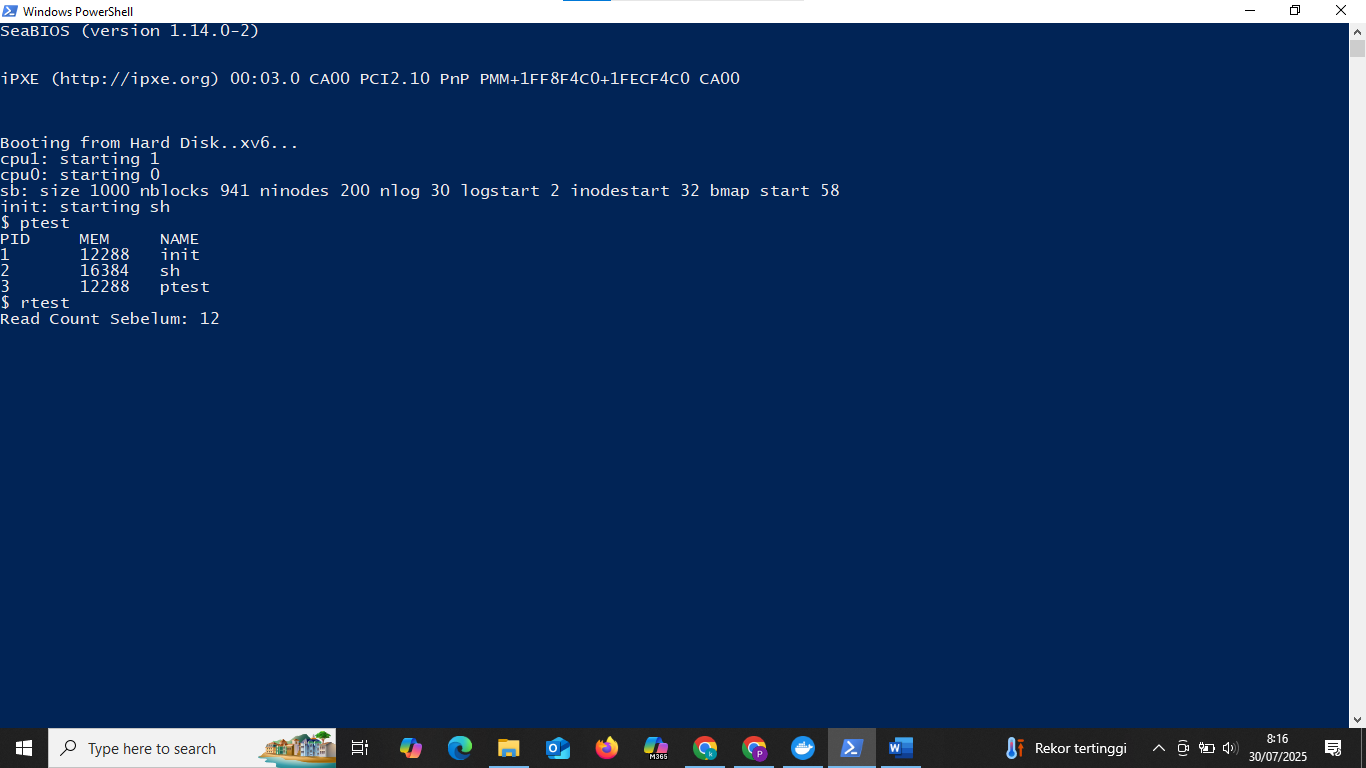
3 12288 ptest

**📍 Output rtest:**

$ rtest

Read Count Sebelum: 12

*Jika disertakan screenshot, dapat ditambahkan seperti berikut:*



**⚠️ Kendala yang Dihadapi**

* ❌ **Pointer dari user space error**  
  Awalnya argptr() gagal karena struktur struct pinfo belum lengkap atau belum di-cast dengan benar.
* ❌ **Kernel panic saat sleep()**  
  Terjadi panic karena memanggil sleep() dengan lock yang belum diinisialisasi atau sembarang pointer.
* ⚠️ **Error: array has incomplete element type**  
  Terjadi karena deklarasi array struct pinfo ptable[NPROC] dilakukan sebelum struct pinfo lengkap didefinisikan.
* ❌ **Kesalahan dalam indexing array**  
  Lupa membuat indeks eksplisit i++ saat mengisi array pinfo, sehingga hanya satu proses yang tercetak.

Solusi dilakukan dengan:

* Memastikan pointer valid dan benar-benar mengacu ke alamat user space yang telah dialokasikan.
* Menempatkan definisi struct pinfo sebelum digunakan.
* Menggunakan loop dan indexing yang benar saat mengisi array dari kernel.

**📚 Referensi**

* Buku xv6 (MIT): <https://pdos.csail.mit.edu/6.828/2018/xv6/book-rev11.pdf>
* Repositori xv6-public: <https://github.com/mit-pdos/xv6-public>
* Diskusi GitHub Issues dan Stack Overflow
* Diskusi dengan teman dan asisten praktikum