**Laporan Latihan Praktikum ke-X  
Sistem Operasi**

**Disusun oleh:**

**Antonius Munthe | 121140032**

**Kelas SIAKAD RA | Cluster A**



**Program Studi Teknik Informatika**

**Jurusan Teknologi Produksi dan Industri**

**Institut Teknologi Sumatera**

**Lampung Selatan**

**2023**

**BAB I  
TEORI DASAR**

**1.1 System Call**

System call ialah suatu metode program komputer dalam meminta layanan dari kernel sistem operasi di mana program tersebut dijalankan. Hal tersebut memungkinkan program untuk berinteraksi dengan sistem operasi, dengan program komputer melakukan pemanggilan sistem ketika memerlukan permintaan kepada kernel sistem operasi. Melalui antarmuka Program Aplikasi (API), System call memberikan layanan sistem operasi kepada program pengguna. Dengan adanya antarmuka antara proses dan sistem operasi, pemanggilan sistem memungkinkan proses pengguna untuk meminta layanan dari sistem operasi. Satu-satunya titik masuk ke dalam sistem kernel adalah melalui pemanggilan sistem, sehingga semua program yang membutuhkan sumber daya harus menggunakan metode ini.

**1.2 Fitur System Call**

* **Interface**

Panggilan sistem menyediakan antarmuka yang terdefinisi dengan baik antara

program pengguna dan sistem operasi. Program membuat permintaan dengan

memanggil fungsi tertentu, dan sistem operasi merespons dengan menjalankan

layanan yang diminta dan mengembalikan hasilnya.

* **Protection**

Panggilan sistem digunakan untuk mengakses operasi istimewa yang tidak tersedia

untuk program pengguna normal. Sistem operasi menggunakan hak istimewa ini

untuk melindungi sistem dari akses berbahaya atau tidak sah.

* **Kernel**

Saat panggilan sistem dibuat, program untuk sementara dialihkan dari mode

pengguna ke mode kernel. Dalam mode kernel, program memiliki akses ke semua

sumber daya sistem, termasuk perangkat keras, memori, dan proses lainnya.

* **Context Switching**

Panggilan sistem memerlukan pengalihan konteks, yang melibatkan penyimpanan

status proses saat ini dan beralih ke mode kernel untuk menjalankan layanan yang

diminta. Ini dapat menimbulkan biaya tambahan, yang dapat memengaruhi kinerja

sistem.

* **Error Handling**

Panggilan sistem dapat mengembalikan kode kesalahan untuk menunjukkan masalah

dengan layanan yang diminta. Program harus memeriksa kesalahan ini dan

menanganinya dengan tepat.

**1.3 Jenis System Call**

* **Process Control System Calls**

System call ini digunakan untuk mengendalikan proses dalam sistem operasi, seperti

memulai proses baru, menghentikan proses yang sedang berjalan, dan melakukan

sinkronisasi antara proses.

* **File Management System Calls**

System call ini digunakan untuk mengelola file dan direktori dalam sistem operasi, seperti membuka, menutup, membaca, menulis, dan memodifikasi file.

* **Device Management System Calls**

System call ini digunakan untuk mengelola perangkat keras dalam sistem operasi,

seperti memulai dan menghentikan perangkat keras, membaca dan menulis data dari

perangkat keras, dan mengubah status perangkat keras.

* **Information Maintenance System Calls**

System call ini digunakan untuk mengelola informasi dalam sistem operasi, seperti

mendapatkan informasi tentang sistem, waktu, tanggal, dan konfigurasi jaringan.

* **Communication System System Calls**

System call ini digunakan untuk mengelola komunikasi antar proses dan antar

jaringan, seperti membuat soket untuk koneksi jaringan, mengirim dan menerima data

melalui jaringan, dan mengelola protokol jaringan.

* **Memory Management System Calls**

System call ini digunakan untuk mengelola memori dalam sistem operasi, seperti

memperoleh memori baru, menghapus memori yang tidak digunakan lagi, dan

memori maping antara proses dan alamat fisik.

**1.4 Beberapa syntax akan digunakan**

* **O\_RDONLY, O\_WRONLY dan O\_RDWR**

Fungsinya yaitu meminta membuka file hanya-baca, hanya-menulis atau membaca maupun menulis, masing-masing, bitwise- atau dengan nol atau lebih.

* **O\_CREAT**

Sebuah fungsi untuk membuatkan sebuah file yang tidak ada.

* **O\_TRUNC**

Jika file sudah ada dan merupakan file biasa dan mode terbuka memungkinkan penulisan (yaitu, O\_RDWR atau O\_WRONLY), maka akan dipotong menjadi panjang 0. Jika file tersebut adalah file perangkat FIFO atau terminal, bendera O\_TRUNC diabaikan. Jika tidak, efek O\_TRUNC tidak ditentukan.

* **EXIT\_FAILURE**

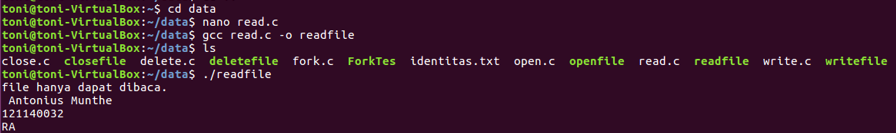
Ini dapat digunakan untuk mengakhiri program secara normal tetapi dengan kondisi operasi gagal. EXIT\_FAILURE memiliki nilai 1.

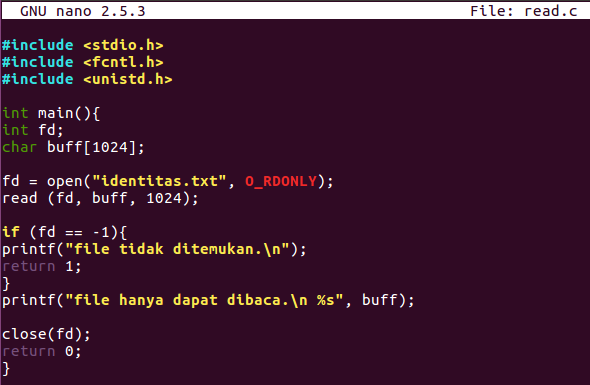
**BAB II  
PEMBAHASAN DAN ANALISIS**

**2.1 Latihan Pertama: File Management dan implementasikan File Management jenis READ**

**2.1.1 Langkah Pertama**

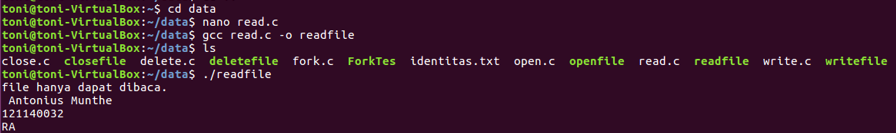
Langkah pertama akses dulu folder data yang sudah dibuat tadi kemudian buat file dengan ekstensi c dengan nama “read.c” kemudian isi dengan kode berikut.





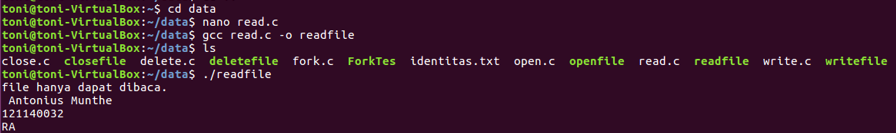
**2.1.2 Langkah Kedua**

Kemudian jalankan perintah “gcc read -o readfile” dan cek apakah readfile berhasil dibuat



**2.1.3 Langkah Ketiga**

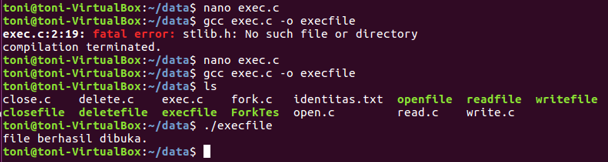
Lakukan pemanggilan sistem yang sudah dibuat tadi dengan perintah “./readfile”



**2.2 Latihan Kedua: Control Proses dan implementasikan Control Proses jenis EXEC**

**2.2.1 Langkah Pertama**

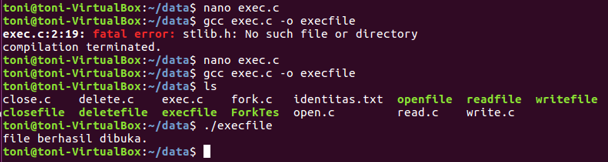
buat file dengan ekstensi c dengan nama “exec.c”.





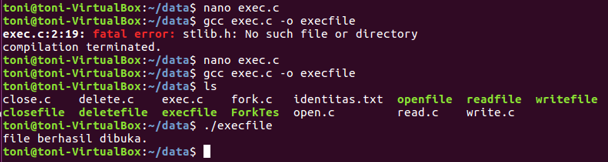
**2.2.2 Langkah Kedua**

Kemudian jalankan perintah “gcc exec -o execfile”



**2.2.3 Langkah Ketiga**

Lakukan pemanggilan sistem yang sudah dibuat tadi dengan perintah “./execfile”



**BAB III  
KESIMPULAN**

Dalam dunia sistem operasi, terdapat sebuah metode yang digunakan oleh program komputer untuk meminta layanan dari kernel sistem operasi, yaitu system call. Beberapa fitur penting dari system call meliputi antarmuka yang terdefinisi dengan baik, perlindungan, mode kernel, pengalihan konteks, dan penanganan kesalahan. Jenis-jenis system call juga bervariasi, antara lain untuk pengendalian proses, pengelolaan file dan direktori, pengelolaan perangkat keras, pemeliharaan informasi, komunikasi antar proses dan antar jaringan, serta pengelolaan memori. Dalam system call, terdapat beberapa syntax umum yang digunakan seperti O\_RDONLY, O\_WRONLY, O\_RDWR, O\_CREAT, O\_TRUNC, dan EXIT\_FAILURE.

**BAB IV  
DAFTAR PUSTAKA**

[1] Modul 3 Praktikum Strategi Algoritma