**Laporan Latihan Praktikum ke-3**

**Sistem Operasi**

**Disusun oleh:**

**Amdhan Anggoro | 121140226**

**Kelas RD**



**Program Studi Teknik Informatika**

**Jurusan Teknik Elektro, Informatika, dan Sistem Fisika**

**Institut Teknologi Sumatera**

**Lampung Selatan**

**2023**

**BAB I  
TEORI DASAR**

**1.1 Teori Dasar 1**

Sistem call adalah memungkinkan program aplikasi untuk berinteraksi dengan sistem operasi dan menggunakan layanan yang disediakan oleh sistem operasi tersebut. Program aplikasi memerlukan layanan sistem operasi, seperti akses ke perangkat keras, memori, manajemen file, manajemen proses, manajemen jaringan, dan sistem keamanan. Sistem call memberikan antarmuka yang terdefinisi dengan baik antara program pengguna dan sistem operasi. Saat program memanggil sistem, program sementara dialihkan dari mode pengguna ke mode kernel, di mana program memiliki akses ke semua sumber daya sistem. Namun, pemanggilan sistem memerlukan pengalihan konteks, yang dapat menimbulkan biaya tambahan dan memengaruhi kinerja sistem.

**1.2 Teori Dasar 2**

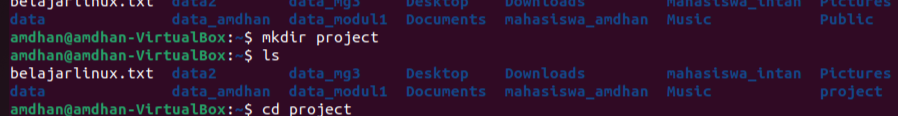
Sistem call adalah metode program komputer dalam meminta layanan dari kernel sistem operasi. Sistem call menyediakan antarmuka yang terdefinisi dengan baik antara program pengguna dan sistem operasi, dengan program membuat permintaan dengan memanggil fungsi tertentu, dan sistem operasi merespons dengan menjalankan layanan yang diminta dan mengembalikan hasilnya.

**BAB II  
PEMBAHASAN DAN ANALISIS**

**2.1 Latihan Pertama**

**2.1.1 Langkah Pertama**

Buatlah sebuah folder atau direktori dengan nama **“Project”** lalu masuk ke direktori **Project.**

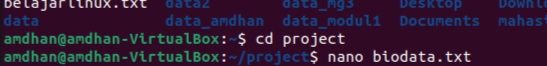


**2.1.2 Langkah Kedua**

Buat sebuah file **“biodata.txt”** dengan isi sebagai berikut:

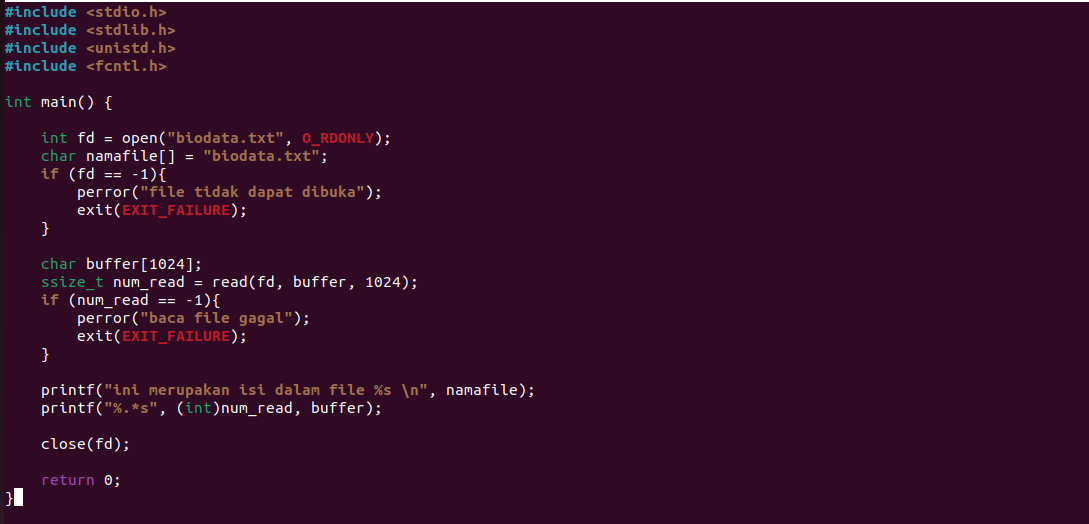


Perintah yang dipakai untuk membuat biodata adalah “**nano**”



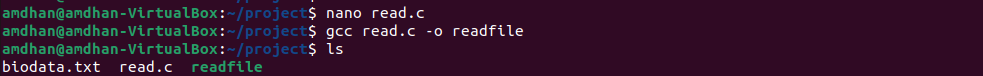
**2.1.3 Langkah Ketiga**

Buatlah sebuah file dengan extensi C dengan nama **“read.c”** pada folder **project** dengan isi sebagai berikut.



**2.1.4 Langkah Keempat**

Jalankan perintah **“gcc read.c -o readfile”.** Lalu periksa isi file



**2.1.5 Langkah Kelima**

jalan kan file **“readfile”** dengan perintah **“./readfile”**



**2. Latihan Kedua**

**2.2.1 Langkah Pertama**

Buatlah sebuah file dengan extensi C dengan nama **“exec.c”** pada folder **project** dengan isi sebagai berikut.



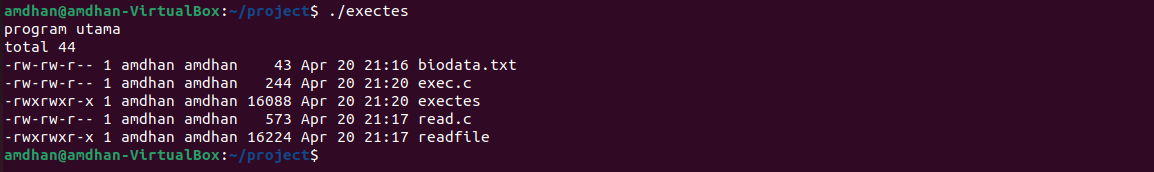
**2.2.2 Langkah Kedua**

Jalankan perintah **“gcc exec.c -o exectes”.** Lalu periksa isi file



**2.2.3 Langkah Ketiga**

jalan kan file **“exectes”** dengan perintah **“./exectes”**



**BAB III  
KESIMPULAN**

Adapun beberapa perintah atau fungsi yang digunakan untuk praktikum kali ini, diantara lain:

* int fd = open("identitas.txt", O\_RDONLY); digunakan Membuka file bernama "identitas.txt" dengan mode hanya membaca (O\_RDONLY) dan menyimpan file descriptor (fd) ke dalam variabel fd. Jika pembukaan file gagal, program akan menampilkan pesan error dan keluar dari program menggunakan fungsi exit().
* char buffer[1024]; ini merupakan variabel array karakter yang akan digunakan untuk menyimpan isi file yang akan dibaca.
* ssize\_t num\_read = read(fd, buffer, 1024); digunakan untuk Membaca isi file dengan menggunakan fungsi read(). Fungsi ini akan mengembalikan jumlah byte yang berhasil dibaca, atau -1 jika terjadi error.
* printf("ini merupakan isi dalam file %s \n", namafile); Menampilkan pesan yang terdapat didalam namafile.