**Laporan Latihan Praktikum ke-3  
Sistem Operasi**

**Disusun oleh:**

**Elika Eugenia Ramadhania | 121140212**

**Kelas RD**



**Program Studi Teknik Informatika**

**Jurusan Teknologi Produksi dan Industri**

**Institut Teknologi Sumatera**

**Lampung Selatan**

**2023**

**BAB I  
TEORI DASAR**

**1.1 Teori Dasar 1**

**LINUX**

Linux adalah sistem operasi yang bersifat multi user, multi tasking yang berbasis UNIX [3], yang merupakan implementasi independen dari standard IEEE untuk OS yang bernama POSIX (Portable Operating System Interface). OS adalah perangkat lunak (software) yang mengatur koordinasi kerja antar semua perlengkapan perangkat keras (hardware) dalam sebuah komputer. Linux memiliki kemampuan yang berbasis ke standard POSIX meliputi truemultitasking, virtual memory, shared libraries, demand-loading, proper memory management, dan multiuser. Linux seperti layaknya OS UNIX lainnnya, mendukung banyak software mulai dari TEX, X Windows, GNU C/C++ sampa ke TCP/IP. Linux adalah sistem operasi yang disebarkan secara luas dengan bebas dibawah lisensi GNU General Public License (GPL), yang berarti jufa source code Linux tersedia.

Kernel yang digunakan adalah sama-sama Linux kernel, sedangkan perbedaannya hanyalah paket-paket aplikasi yang disertakan, sistem penyusunan direktori, init style, dll. Linux sekarang merupakan alternatif OS yang jauh lebih murah jika dibandingkan dengan OS komersial, dengan kemampuan yang setara atau bahkan lebih. Linux bukan lagi suatu OS turunan Unix yang text based, tapi 7 sudah berubah menjadi OS yang memiliki GUI yang lebih indah dan fleksible daripada Microsoft Windows.

**Virtual Machine**

Mesin Virtual pada mulanya didefinisikan oleh Gerard J. Popek dan Robert P. Goldberg pada tahun 1974. Mesin Virtual machine tersebut merupakan sebuah duplikat efektif yang tertutup dari mesin yang asli. Pada era modern ini mesin virtual dapat meniru serta menggambarkan secara spesifik dari sesuatu yang nyata meskipun tanpa perangkat asli. Mesin virtual system merupakan sebuah perangkat berupa system platform dimana perangkat tersebut berfungsi sebagai pengoperasian sebuah sistem.

Sedangkan mesin virtual proses diciptakan sebagai perintah dalam pengoperasian program tertentu pada komputer, dimana mesin tersebut memiliki batasan kemampuannya. Contoh Virtual machine :

1) Microsoft Virtual PC

2) Microsoft sebagai pendia OS Yang terbesar juga mempunyai software Virtual komputer.

3) VMware VMware merupakan sebuah perangkat lunak yang dapat dijalankan pada Microsoft Windows, linux dan Mac OSX secara langsung tanpa perlu tambahan dasar sistem operasi.

4) Oracle VM VirtualBox Oracle VM VirtualBox adalah perangkat lunak yang biasa di fungsionalkan dalam pengeksekusian tambahan pada sistem operasi utama.

**1.2 Teori Dasar 2**

**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.

**Fitur System Call**

1. Interface

Panggilan sistem menyediakan antarmuka yang terdefinisi dengan baik antara program pengguna dan sistem operasi.

1. Protection

Panggilan sistem digunakan untuk mengakses operasi istimewa yang tidak tersedia untuk program pengguna normal.

1. Kernel

Dalam mode kernel, program memiliki akses ke semua sumber daya sistem, termasuk perangkat keras, memori, dan proses lainnya.

1. Context Switching

Panggilan sistem memerlukan pengalihan konteks, yang melibatkan penyimpanan status proses saat ini dan beralih ke mode kernel untuk menjalankan layanan yang diminta.

1. Error Handling

Program harus memeriksa kesalahan ini dan menanganinya dengan tepat.

1. Context Switching

Panggilan sistem dapat digunakan untuk menyinkronkan akses ke sumber daya bersama, seperti file atau koneksi jaringan.

**Jenis System Call**

1. 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.

1. File Management System Calls

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

1. 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.

1. Information Maintenance System Calls

System call ini digunakan untuk mengelola informasi dalam sistem operasi, seperti mendapatkan informasi tentang sistem, waktu, tanggal, dan konfigurasi jaringan.

1. 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.

1. Memory Management System Calls

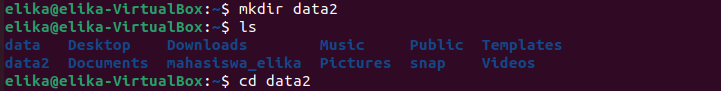
System call ini digunakan untuk mengelola memori dalam sistem operasi, seperti memperoleh memori baru, menghapus memori yang tidak digunakan lagi, dan memori maping antara proses dan alamat fisik.

**BAB II  
PEMBAHASAN DAN ANALISIS**

**2.1 Latihan Pertama : Pengimplementasian File Management jenis READ**

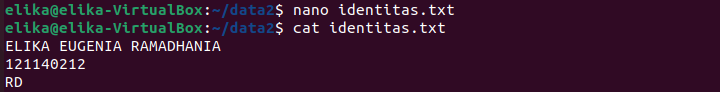
**2.1.1 Langkah Pertama**

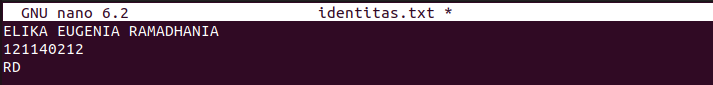
Membuat sebuah folder baru bernama ‘data2’ menggunakan perintah mkdir dan menggunakan folder tersebut dengan perintah ‘cd data2’.



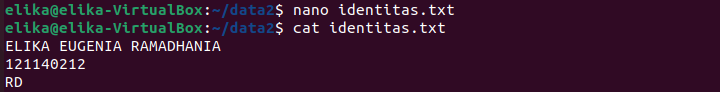
**2.1.2 Langkah Kedua**

Membuat sebuah file bernama ‘identitas.txt’ yang berisi nama, nim, dan kelas menggunakan perintah ‘nano.identitas.txt’.





untuk menampilkan isi dari file tersebut menggunakan perintah ‘cat identitas.txt’.



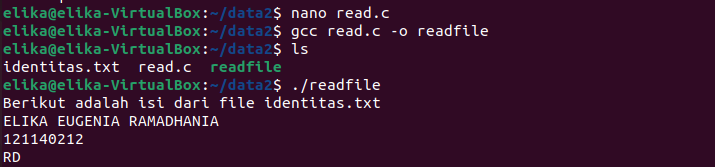
**2.1.3 Langkah Ketiga**

Membuat sebuah file dengan extensi C dengan nama ‘read.c’ pada folder data2 dengan isi sebagai berikut.



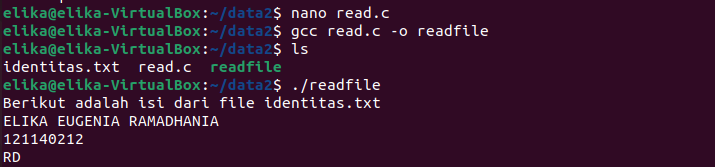
**2.1.4 Langkah Keempat**

Menggunakan perintah ‘gcc read.c -o readfile’ dan memeriksa isi file dengan perintah ‘ls’.



**2.1.4 Langkah Kelima**

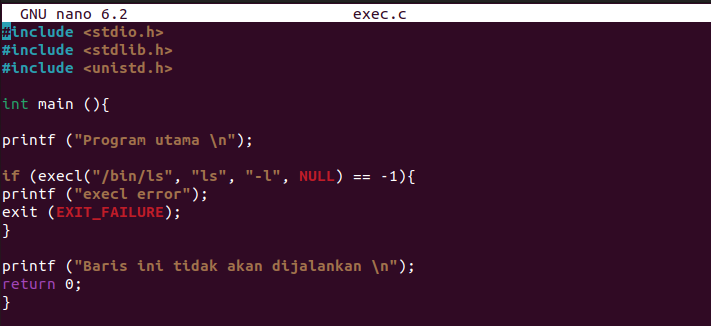
Menjalankan file menggunakan perintah ‘./readfile’, maka akan muncul seperti pada gambar dibawah ini.



**2.2 Latihan Kedua**

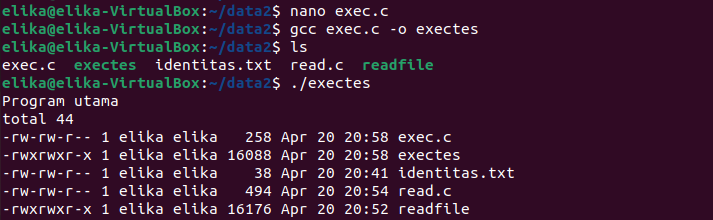
**2.2.1 Langkah Pertama**

Membuat sebuah file dengan extensi C dengan nama ‘exec.c’ pada folder data2 dengan isi sebagai berikut.



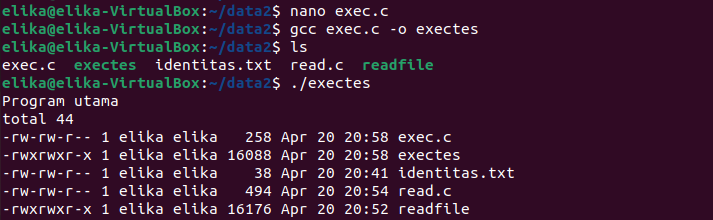
**2.2.2 Langkah Kedua**

Menggunakan perintah ‘gcc exec.c -o exectes’ dan memeriksa isi file dengan perintah ‘ls’.



**2.2.3 Langkah Ketiga**

Menjalankan file menggunakan perintah ‘./exectes’, maka akan muncul seperti pada gambar dibawah ini.



**BAB III  
KESIMPULAN**

Pada bab ini dijabarkan beberapa kesimpulan terkait kegunaan perintah yang digunakan pada praktikum:

1. Membuka file, gcc open.c -o openfile
2. Menutup file, gcc close.c -o closefile
3. Menulis file, gcc write.c -o writefile
4. Menghapus file, gcc delete.c -o deletefile
5. Implementasi system call fork, gcc fork.c -o ForkTes
6. Implementasi system call wait, gcc fork.c -o WaitTes