

Laporan Latihan Praktikum ke-2

Sistem Operasi

Disusun oleh:
Indah Mutiara | 121140158
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 “data” lalu pindah ke direktori data2.

```
indahmutiara@indahmutiara-VirtualBox:~$ ls
Desktop  Downloads  Music      Public  Templates
Documents mahasiswa_indah Pictures snap    Videos
indahmutiara@indahmutiara-VirtualBox:~$ mkdir data
indahmutiara@indahmutiara-VirtualBox:~$ ls
data      Documents mahasiswa_indah Pictures snap    Videos
Desktop  Downloads Music      Public  Templates
indahmutiara@indahmutiara-VirtualBox:~$ cd data
```

2.1.2 Langkah Kedua

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

NAMA : INDAH MUTIARA

NIM : 121140158

KELAS : RD

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano datadiri.txt
indahmutiara@indahmutiara-VirtualBox:~/data$ cat datadiri.txt
NAMA : INDAH MUTIARA
NIM : 121140158
KELAS : RD
```

2.1.3 Langkah Ketiga

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

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano read.c

GNU nano 6.2 read.c *
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main(){
//openfile
int fd = open("datadiri.txt", O_RDONLY);
char namafile[] = "datadiri.txt";
if(fd == -1){
perror("File tidak bisa dibuka");
exit(EXIT_FAILURE);
}
//readfile
char buffer[1024];
ssize_t num_read = read(fd, buffer, 1024);
if (num_read == -1){
perror("File tidak bisa dibaca");
exit(EXIT_FAILURE);
}
//showfile
printf("ini isi filenya %s \n", namafile);
printf("%.s", (int)num_read, buffer);
//closefile
close(fd);

return 0;
}
```

| | | | | | |
|---------|--------------|-------------|----------|------------|---------------|
| ^G Help | ^O Write Out | ^W Where Is | ^K Cut | ^T Execute | ^C Location |
| ^X Exit | ^R Read File | ^_ Replace | ^U Paste | ^J Justify | ^_ Go To Line |

2.1.4 Langkah Keempat

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

```
indahmutiara@indahmutiara-VirtualBox:~/data$ gcc read.c -o readfile
indahmutiara@indahmutiara-VirtualBox:~/data$ ls
datadiri.txt  read.c  readfile
```

2.1.5 Langkah Kelima

Jalankan file “readfile” dengan perintah “./readfile”

```
indahmutiara@indahmutiara-VirtualBox:~/data$ ls
datadiri.txt  read.c  readfile
indahmutiara@indahmutiara-VirtualBox:~/data$ ./readfile
ini isi filenya datadiri.txt
NAMA  : INDAH MUTIARA
NIM   : 121140158
KELAS : RD
indahmutiara@indahmutiara-VirtualBox:~/data$
```

2. Latihan Kedua

2.2.1 Langkah Pertama

Buatlah sebuah file dengan ekstensi C dengan nama “exec.c” pada folder data dengan isi sebagai berikut.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano exec.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
int main(){
printf("Program utama \n");
//jalankan program ls dengan menggunakan exec
if (execl("/bin/ls", "ls", "-l", NULL) == -1){
perror("exec error");
exit(EXIT_FAILURE);
}
printf("baris ini tidak dijalankan \n");
return 0;
}
```

2.2.2 Langkah Kedua

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

```
indahmutiara@indahmutiara-VirtualBox:~/data$ gcc exec.c -o exectes
indahmutiara@indahmutiara-VirtualBox:~/data$ ls
datadiri.txt  exec.c  exectes  read.c  readfile
```

2.2.3 Langkah Ketiga

Jalankan file “exectes” dengan perintah “./exectes”

```
indahmutiara@indahmutiara-VirtualBox:~/data$ ./exectes
Program utama
total 44
-rw-rw-r-- 1 indahmutiara indahmutiara  51 Apr 20 19:28 datadiri.txt
-rw-rw-r-- 1 indahmutiara indahmutiara   289 Apr 20 20:02 exec.c
-rwxrwxr-x 1 indahmutiara indahmutiara 16088 Apr 20 20:03 exectes
-rw-rw-r-- 1 indahmutiara indahmutiara   526 Apr 20 19:41 read.c
-rwxrwxr-x 1 indahmutiara indahmutiara 16224 Apr 20 19:52 readfile
indahmutiara@indahmutiara-VirtualBox:~/data$
```

BAB III

KESIMPULAN

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

- `char buffer[1024];` merupakan 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.