LAPORAN PRAKTIKUM
SISTEM OPERASI 2024
MODUL 3

Nama : **Aulia Putri Sayidina**

NIM : **122140060**

Kelas : **RB**

PROGRAM STUDI TEKNIK INFORMATIKA
INSTITUT TEKNOLOGI SUMATERA
2024

DAFTAR ISI

1. DASAR TEORI

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.

2. System Call

2.1 Menerapkan System Call jenis “READ”

System call read digunakan untuk membaca data dari suatu file descriptor atau input standar. Data yang dibaca akan disimpan di dalam buffer yang disediakan oleh pengguna.

1. Buat file dengan ekstensi C dengan nama ”read.c” dan file dengan code untuk mempraktikan contoh system call jenis READ.

```
Vbox@aul:~/data$ nano read.c
Vbox@aul:~/data$ cat read.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main() {
    // Buka file untuk dibaca (READ-ONLY)
    int file_descriptor = open("identitas.txt", O_RDONLY);

    // Periksa apakah file berhasil dibuka
    if (file_descriptor == -1) {
        perror("Error opening file");
        return EXIT_FAILURE;
    }

    // Baca isi file
    char buffer[100];  // Buffer untuk menyimpan data yang dibaca
    int bytes_read = read(file_descriptor, buffer, sizeof(buffer));

    // Periksa apakah pembacaan file berhasil
    if (bytes_read == -1) {
        perror("Error reading file");
        close(file_descriptor);  // Tutup file
        return EXIT_FAILURE;
    }

    // Tampilkan isi file yang telah dibaca
    printf("Isi file 'identitas.txt':\n%s\n", buffer);

    // Tutup file setelah selesai membaca
    close(file_descriptor);

    return EXIT_SUCCESS;
}
```

2. Jalankan perintah ”gcc read.c -o readfile” dan run perintah ”./readfile”.

```
Vbox@aul:~/data$ gcc read.c -o readfile
Vbox@aul:~/data$ ls
close.c  closefile  delete.c  deletefile  identitas.txt  open.c  openfile  read.c  readfile  write.c  writefile
Vbox@aul:~/data$ ./readfile
Isi file 'identitas.txt':
Aulia Putri Sayidina
122140060
RB
```

2.2 Menerapkan System Call jenis “EXEC”

System call EXEC digunakan untuk menggantikan proses yang sedang berjalan dengan proses baru yang dieksekusi. Proses baru ini akan mengambil alih proses yang sebelumnya sedang berjalan.

1. Buat file dengan ekstensi C dengan nama ”exec.c” dan file dengan code untuk mempraktikan contoh system call jenis EXEC.

```
Vbox@aul:~/data$ nano exec.c
Vbox@aul:~/data$ cat exec.c
#include <stdio.h>
#include <stdlib.h>

int main() {
    // Mengeksekusi perintah/command menggunakan system()
    printf("Mengeksekusi perintah 'ls -l':\n");
    system("ls -l");

    return EXIT_SUCCESS;
}
```

2. Jalankan perintah ”gcc exec.c -o exec” dan run perintah ”./exec”.

```
Vbox@aul:~/data$ gcc exec.c -o exec
Vbox@aul:~/data$ ls
close.c  closefile  delete.c  deletefile  exec  exec.c  identitas.txt  open.c  openfile  read.c  readfile  write.c  writefile
Vbox@aul:~/data$ ./exec
Mengeksekusi perintah 'ls -l':
total 124
-rw-rw-r-- 1 wupxy wupxy   270 Apr 18 09:28 close.c
-rwxrwxr-x 1 wupxy wupxy 16040 Apr 18 09:29 closefile
-rw-rw-r-- 1 wupxy wupxy   287 Apr 18 09:45 delete.c
-rwxrwxr-x 1 wupxy wupxy 16056 Apr 18 09:46 deletefile
-rwxrwxr-x 1 wupxy wupxy 16000 Apr 20 22:36 exec
-rw-rw-r-- 1 wupxy wupxy   208 Apr 20 22:36 exec.c
-rw-rw-r-- 1 wupxy wupxy    34 Apr 20 22:10 identitas.txt
-rw-rw-r-- 1 wupxy wupxy   245 Apr 18 09:23 open.c
-rwxrwxr-x 1 wupxy wupxy 16040 Apr 18 09:24 openfile
-rw-rw-r-- 1 wupxy wupxy   893 Apr 20 22:34 read.c
-rwxrwxr-x 1 wupxy wupxy 16176 Apr 20 22:34 readfile
-rw-rw-r-- 1 wupxy wupxy   433 Apr 18 09:39 write.c
-rwxrwxr-x 1 wupxy wupxy 16224 Apr 18 09:40 writefile
Vbox@aul:~/data$ S
```

3. KESIMPULAN

System Call adalah antarmuka antara program aplikasi dan sistem operasi. Melalui System Call, program dapat berinteraksi dengan berbagai layanan yang disediakan oleh sistem operasi, seperti membaca atau menulis ke dalam file, mengelola proses, mengalokasikan memori, dan lainnya.