LAPORAN PRAKTIKUM
SISTEM OPERASI RD
MODUL 3

Oleh :

Rayhan Fadel Irwanto (122140236)

Program Studi Teknik Informatika

Institut Teknologi Sumatera

2024

Daftar Isi

Daftar Isi 2
1. Dasar Teori 3
2. Hasil & Jawaban 3
3. Kesimpulan dan Saran 5

1. Dasar Teori

System Call **READ** digunakan untuk membaca data dari file descriptor atau input standar, dimana data yang dibaca akan disimpan di dalam buffer yang telah dialokasikan. Sedangkan System Call **EXEC** digunakan untuk menggantikan proses yang sedang berjalan dengan proses baru yang akan dieksekusi, sehingga proses baru tersebut mengambil alih jalannya proses yang sebelumnya berjalan.

2. Hasil & Jawaban

Percobaan 1 READ

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

```
Vbox@rehan:~/data1$ nano testfileread.txt
Vbox@rehan:~/data1$ ls
fork.c  forkTes  testfileread.txt  wait.c  waitTes
Vbox@rehan:~/data1$ nano read.c
Vbox@rehan:~/data1$ nano read.c
Vbox@rehan:~/data1$ rm read.c
Vbox@rehan:~/data1$ nano read.c
Vbox@rehan:~/data1$ cat read.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <fcntl.h>

int main() {
    int file_descriptor = open("testfileread.txt", O_RDONLY);

    if (file_descriptor == -1) {
        printf("Gagal membuka file");
        return EXIT_FAILURE;
    }

    char buffer[100];
    int bytes_read = read(file_descriptor, buffer, sizeof(buffer));

    if (bytes_read == -1) {
        printf("Gagal membaca file");
        close(file_descriptor);
        return EXIT_FAILURE;
    }

    printf("Isi 'testfileread.txt':\n%s\n", buffer);

    close(file_descriptor);

    return EXIT_SUCCESS;
}
```

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

```
Vbox@rehan:~/data1$ gcc read.c -o readFile
Vbox@rehan:~/data1$ ./readFile
Isi 'testfileread.txt':
ini merupakan isi dari file read
```

Perobaan 2 EXEC

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

```
Vbox@rehan:~/data1$ nano exec.c
Vbox@rehan:~/data1$ cat exec.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main() {
    // Menggunakan system call exec untuk mengeksekusi perintah "ls -l"
    printf("Mengeksekusi perintah 'ls -l':\n");

    // Menyiapkan argumen untuk perintah exec
    char *args[] = {"ls", "-l", NULL};

    // Mengeksekusi perintah menggunakan execvp
    if (execvp("ls", args) == -1) {
        perror("Gagal mengeksekusi perintah");
        return EXIT_FAILURE;
    }

    return EXIT_SUCCESS;
}
```

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

```
Vbox@rehan:~/data1$ gcc exec.c -o execFile
Vbox@rehan:~/data1$ ./execFile
Mengeksekusi perintah 'ls -l':
total 84
-rw-rw-r-- 1 wupxy wupxy   474 Apr 23 22:47 exec.c
-rwxrwxr-x 1 wupxy wupxy 16096 Apr 23 22:47 execFile
-rw-rw-r-- 1 wupxy wupxy   229 Apr 23 22:39 fork.c
-rwxrwxr-x 1 wupxy wupxy 16040 Apr 23 22:39 forkTes
-rw-rw-r-- 1 wupxy wupxy   597 Apr 23 22:45 read.c
-rwxrwxr-x 1 wupxy wupxy 16136 Apr 23 22:46 readFile
-rw-rw-r-- 1 wupxy wupxy    33 Apr 23 22:41 testfileread.txt
-rw-rw-r-- 1 wupxy wupxy   308 Apr 23 22:40 wait.c
-rwxrwxr-x 1 wupxy wupxy 16176 Apr 23 22:40 waitTes
Vbox@rehan:~/data1$
```

3. Kesimpulan dan Saran

Setelah menjalani praktikum yang menitikberatkan pada System Call READ dan EXEC, dapat dipahami bahwa keduanya memiliki peran krusial dalam hubungan antara program pengguna dan kernel sistem operasi. System Call READ memfasilitasi proses membaca data dari file atau input device, sementara System Call EXEC digunakan untuk menginisiasi eksekusi program baru dalam konteks proses yang tengah berjalan.