

Laporan Percobaan 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

1. Percobaan Pertama : Membuka File dengan system call

1.1 Langkah Pertama

Membuat folder “Data” dan mengakses isi folder tersebut.

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

1.2 Langkah Kedua

Membuat file dengan nama “identitas.txt” dengan isi Nama, Nim, Kelas.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano identitas.txt
indahmutiara@indahmutiara-VirtualBox:~/data$ cat identitas.txt
Indah mutiara
121140158
RD
```

1.3 Langkah Ketiga

Membuat file dengan ekstensi C dengan nama “open.c”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano open.c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>

int main(){
int fd;
fd = open("identitas.txt", O_RDWR);
if(fd == -1){
printf("file tidak ditemukan. \n");
return 1;
}
printf("file berhasil dibuka. \n");
close(fd);
return 0;
}
```

1.4 Langkah Keempat

Bila telah melakukan instalasi **gcc** dengan perintah “**sudo apt install gcc**”, maka jalankan perintah “**gcc open.c -o openfile**”.

```
sudo apt install gcc
indahmutiara@indahmutiara-VirtualBox:~/data$ gcc open.c -o openfile
```

1.5 Langkah Kelima

Lakukan pemanggilan sistem yang telah kita buat dengan perintah “**./openfile**”. Jika file berhasil dibuka maka akan menampilkan luaran berikut.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ ./openfile
file berhasil dibuka.
indahmutiara@indahmutiara-VirtualBox:~/data$
```

2. Percobaan Kedua : Menutup File dengan System Call.

2.1 Langka Pertama

Mengakses folder “Data” dan membuaf file dengan ekstensi C dengan nama “close.c”

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano close.c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>

int main(){
int fd;
fd = open("identitas.txt",O_RDONLY);

if (fd == -1){
printf("file tidak ditemukan. \n");
return 1;
}
printf("file berhasil dibuka. \n");

close(fd);
printf("file berhasil ditutup. \n");
return 0;
}
```

2.2 Langkah Kedua

Jalankan perintah “**gcc close.c -o closefile**”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ gcc close.c -o closefile
indahmutiara@indahmutiara-VirtualBox:~/data$ ls
close.c  closefile  identitas.txt  open.c  openfile
```

2.3 Langkah Ketiga

Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./closefile”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ ./closefile
file berhasil dibuka.
file berhasil ditutup.
indahmutiara@indahmutiara-VirtualBox:~/data$
```

3. Percobaan Ketiga : Menulis ke dalam file dengan system call

3.1 Langkah Pertama

Mengakses folder “Data” dan membuat file dengan ekstensi C dengan nama “write.c”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano write.c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>
#include <string.h>

int main(){
    int fd;
    char buf[1024];

    fd = open("contoh.txt", O_CREAT | O_WRONLY | O_TRUNC);
    if(fd == -1){
        printf("error ketika membuka file.\n");
        exit(EXIT_FAILURE);
    }

    strcpy(buf, " Saya sedang melakukan praktikum mata kuliah sistem operasi. \n");
    write(fd, buf, strlen(buf));

    printf("file berhasil ditulis. \n");
    close(fd);
    return 0;
}
```

3.2 Langkah Kedua

Jalankan perintah “gcc write.c -o writefile”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ gcc write.c -o writefile
indahmutiara@indahmutiara-VirtualBox:~/data$ ls
close.c  closefile  identitas.txt  open.c  openfile  write.c  writefile
```

3.3 Langkah Ketiga

Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./writefile”. Jika file berhasil ditulis maka akan menghasilkan file “contoh.txt”

```
indahmutiara@indahmutiara-VirtualBox:~/data$ ./writefile
file berhasil ditulis.
indahmutiara@indahmutiara-VirtualBox:~/data$ cat contoh.txt
cat: contoh.txt: Permission denied
indahmutiara@indahmutiara-VirtualBox:~/data$ ls -l
total 68
-rw-rw-r-- 1 indahmutiara indahmutiara 278 Apr 20 20:31 close.c
-rwxrwxr-x 1 indahmutiara indahmutiara 16040 Apr 20 20:31 closefile
--w--wx--T 1 indahmutiara indahmutiara 62 Apr 20 20:45 contoh.txt
-rw-rw-r-- 1 indahmutiara indahmutiara 27 Apr 20 20:08 identitas.txt
-rw-rw-r-- 1 indahmutiara indahmutiara 236 Apr 20 20:14 open.c
-rwxrwxr-x 1 indahmutiara indahmutiara 16040 Apr 20 20:15 openfile
-rw-rw-r-- 1 indahmutiara indahmutiara 438 Apr 20 20:44 write.c
-rwxrwxr-x 1 indahmutiara indahmutiara 16224 Apr 20 20:44 writefile
indahmutiara@indahmutiara-VirtualBox:~/data$ chmod -R 777 contoh.txt
indahmutiara@indahmutiara-VirtualBox:~/data$ ls -l
total 68
-rw-rw-r-- 1 indahmutiara indahmutiara 278 Apr 20 20:31 close.c
-rwxrwxr-x 1 indahmutiara indahmutiara 16040 Apr 20 20:31 closefile
-rwxrwxrwx 1 indahmutiara indahmutiara 62 Apr 20 20:45 contoh.txt
-rw-rw-r-- 1 indahmutiara indahmutiara 27 Apr 20 20:08 identitas.txt
-rw-rw-r-- 1 indahmutiara indahmutiara 236 Apr 20 20:14 open.c
-rwxrwxr-x 1 indahmutiara indahmutiara 16040 Apr 20 20:15 openfile
-rw-rw-r-- 1 indahmutiara indahmutiara 438 Apr 20 20:44 write.c
-rwxrwxr-x 1 indahmutiara indahmutiara 16224 Apr 20 20:44 writefile
indahmutiara@indahmutiara-VirtualBox:~/data$ cat contoh.txt
Saya sedang melakukan praktikum mata kuliah sistem operasi.
indahmutiara@indahmutiara-VirtualBox:~/data$
```

4. Percobaan Keempat: Menulis ke dalam file dengan system call

4.1 Langkah Pertama

Mengakses folder “Data” dan membuaf file dengan ekstensi C dengan nama “delete.c”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano delete.c
#include <stdio.h>
#include <fcntl.h>
#include <unistd.h>
#include <stdlib.h>

int main(){
char filename[] = "contoh.txt";
int result = unlink(filename);

if(result == 0){
printf("file %s berhasil dihapus. \n",filename);
}else{
printf("gagal menghapus file %s. \n",filename);
}

return 0;
}
```

4.2 Langkah Kedua

Jalankan perintah “gcc delete.c -o deletefile”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ gcc delete.c -o deletefile
indahmutiara@indahmutiara-VirtualBox:~/data$ ls
close.c  contoh.txt  deletefile  open.c  write.c
closefile delete.c  identitas.txt openfile writefile
```

4.3 Langkah Ketiga

Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./deletefile”. Jika file berhasil dihapus maka akan menghasilkan luaran.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ ./deletefile
file contoh.txt berhasil dihapus.
indahmutiara@indahmutiara-VirtualBox:~/data$ ls
close.c  delete.c  identitas.txt  openfile  writefile
closefile deletefile open.c        write.c
indahmutiara@indahmutiara-VirtualBox:~/data$
```

5. Percobaan Kelima : Implementasi System Call Fork

5.1 Langkah Pertama

Mengakses folder “Data1” dan membuat file dengan ekstensi C dengan nama “fork.c”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano fork.c
#include <stdio.h>
#include <unistd.h>

int main(){
    pid_t pid;
    pid = fork();

    if (pid == 1){
        printf ("Fork gagal");
    } else if (pid == 0){
        printf ("Ini adalah proses child .\n");
    } else {
        printf ("Ini adalah proses parent. \n");
    }

    return 0;
}
```

5.2 Langkah Kedua

Jalankan perintah “gcc fork.c -o ForkTes”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ gcc fork.c -o ForkTes
indahmutiara@indahmutiara-VirtualBox:~/data$ ls
close.c  delete.c  fork.c  identitas.txt  openfile  writefile
closefile  deletefile  ForkTes  open.c        write.c
```

5.3 Langkah Ketiga

Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./Fork.Test”. Jika file berhasil dihapus maka akan menghasilkan keluaran

```
indahmutiara@indahmutiara-VirtualBox:~/data$ ./ForkTes
Ini adalah proses parent.
indahmutiara@indahmutiara-VirtualBox:~/data$ Ini adalah proses child .
```


6. Percobaan Keenam : Implementasi System Call Fork

6.1 Langkah Pertama

- . Mengakses folder “Data” dan membuaf file dengan ekstensi C dengan nama “wait.c”

```
indahmutiara@indahmutiara-VirtualBox:~/data$ nano wait.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>

int main(){
pid_t pid;
int status;

pid = fork();

if(pid == 1){
printf("Fork Gagal");
}else if (pid == 0){
printf("Ini adalah proses child. \n");
exit(0);
}else{
wait(&status);
printf("Ini adalah proses parent. \n");
}

return 0;
}
```

6.2 Langkah Kedua

Jalankan perintah “gcc wait.c -o WaitTes”.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ gcc wait.c -o WaitTes
indahmutiara@indahmutiara-VirtualBox:~/data$ ls
close.c    delete.c   fork.c     identitas.txt  openfile  WaitTes  writefile
closefile  deletefile ForkTes    open.c         wait.c    write.c
```

6.3 Langkah Ketiga

Lakukan pemanggilan sistem yang telah kita buat dengan perintah “./WaitTes”. Jika file berhasil dihapus maka akan menghasilkan luaran.

```
indahmutiara@indahmutiara-VirtualBox:~/data$ ./WaitTes
Ini adalah proses child.
Ini adalah proses parent.
indahmutiara@indahmutiara-VirtualBox:~/data$
```