

PRAKTIKUM SISTEM OPERASI

MODUL 9 : FILE SYSTEM CALL



Disusun oleh:

AFIFAH GH AISANI IMANA

L200190198

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

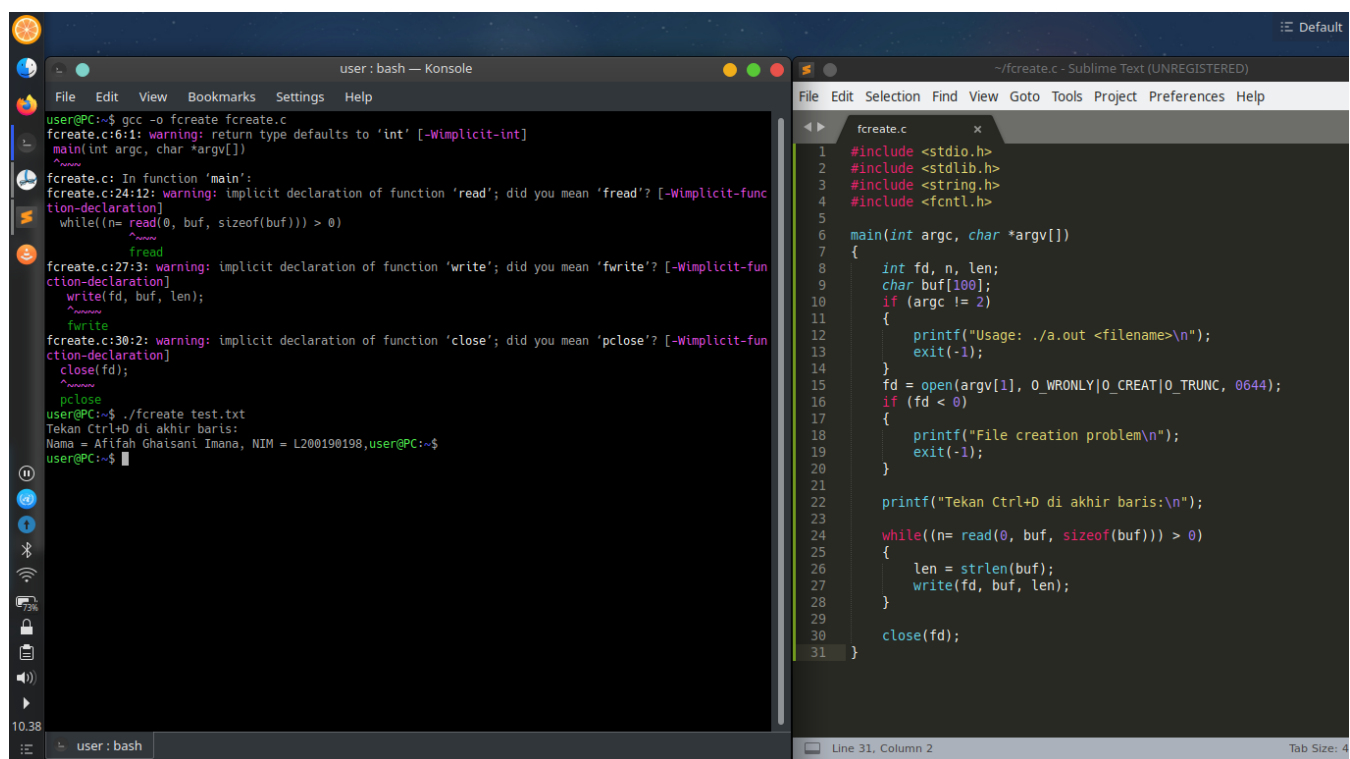
TAHUN 2019/2020

Langkah Kerja.

1. Membuat sebuah file dan menuliskan data di dalamnya.

- ✓ Deklarasi variabel untuk buffer 'buf' untuk menyimpan 100 byte data.
- ✓ Gunakan nama file yang diberikan sebagai argumen pada command-line
- ✓ Buat sebuah file baru dengan nama file seperti pada (2) menggunakan fungsi system-call open() dengan opsi O_CREAT dan O_TRUNC.
- ✓ Periksa 'file-descriptor', jika pembuatan file tidak berhasil, selanjutnya berhenti (stop) dan keluar dari program.
- ✓ Baca input dari console sampai user menekan tombol Ctrl+D. Membaca 100 byte dari data yang dimasukkan melalui console dan menyimpannya ke dalam variabel 'buf' menggunakan perintah system-call read(). Memindahkan isi variabel 'buf' ke dalam file menggunakan perintah 'write'.
- ✓ Menutup file dengan menggunakan fungsi 'close'
- ✓ Stop

Kode Program fcreate.c dan tampilan saat di run pada terminal.



The image shows a terminal window on the left and a code editor window on the right. The terminal window displays the compilation of fcreate.c using gcc, showing several warnings about implicit function declarations for read, write, and close. It then shows the execution of the program, which prompts the user to enter their name and NIM, and then prints the input back to the console.

```
user@PC:~$ gcc -o fcreate fcreate.c
fcreate.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char *argv[])
^~~~~~
fcreate.c: In function 'main':
fcreate.c:24:12: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
while((n= read(0, buf, sizeof(buf))) > 0)
           ^~~~~~
fcreate.c:27:3: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
write(fd, buf, len);
      ^~~~~~
fcreate.c:30:2: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
close(fd);
   ^~~~~~
pclose
user@PC:~$ ./fcreate test.txt
Tekan Ctrl+D di akhir baris:
Nama = Afifah Ghalsani Imana, NIM = L200190190, user@PC:~$
user@PC:~$
```

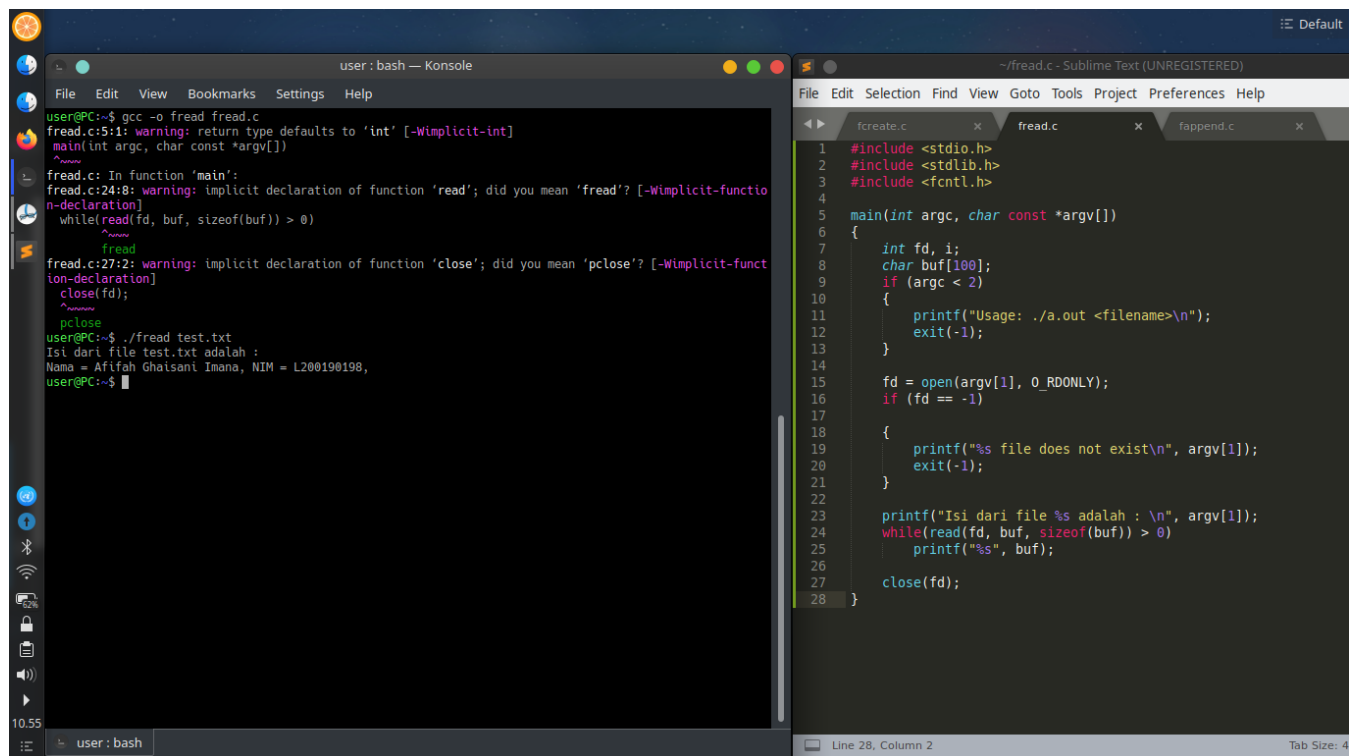
The code editor window shows the source code of fcreate.c, which includes headers for stdio, stdlib, string, and fcntl. The main function takes argc and argv as arguments. It checks if the number of arguments is 2. If not, it prints a usage message and exits. If yes, it opens a file named argv[1] with flags O_WRONLY|O_CREAT|O_TRUNC and mode 0644. If the file opening fails (fd < 0), it prints an error message and exits. Otherwise, it prints a prompt for the user to enter their name and NIM, and then enters a loop where it reads input from the console (0) into a buffer (buf) until Ctrl+D is pressed (n == 0). It then writes the buffer to the file (fd) and closes the file.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 #include <fcntl.h>
5
6 main(int argc, char *argv[])
7 {
8     int fd, n, len;
9     char buf[100];
10    if (argc != 2)
11    {
12        printf("Usage: ./a.out <filename>\n");
13        exit(-1);
14    }
15    fd = open(argv[1], O_WRONLY|O_CREAT|O_TRUNC, 0644);
16    if (fd < 0)
17    {
18        printf("File creation problem\n");
19        exit(-1);
20    }
21
22    printf("Tekan Ctrl+D di akhir baris:\n");
23
24    while((n= read(0, buf, sizeof(buf))) > 0)
25    {
26        len = strlen(buf);
27        write(fd, buf, len);
28    }
29
30    close(fd);
31 }
```

2. Membaca sebuah file dan menampilkan isinya di layar.

- ✓ Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.
- ✓ Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.
- ✓ Buka file untuk dibaca menggunakan perintah 'open' dengan opsi O_RDONLY.
- ✓ Periksa isi file-descriptor, Jika file tidak ada maka program berhenti, stop
- ✓ Baca isi file per 100 byte data menggunakan perintah 'read' sampai ketemu dengan tanda akhir file, 'END-OF-FILE'.
- ✓ Tutup file menggunakan perintah 'close'
- ✓ Stop

Kode Program fread.c dan tampilan saat di run pada terminal.



The image shows a terminal window on the left and a code editor on the right. The terminal window displays the compilation and execution of the program. The code editor shows the source code of the program.

```
user@PC:~$ gcc -o fread fread.c
fread.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char const *argv[])
^~~~~~
fread.c: In function 'main':
fread.c:24:18: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
while(read(fd, buf, sizeof(buf)) > 0)
       ^~~~~~
fread.c:27:2: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
close(fd);
^~~~~~
pclose
user@PC:~$ ./fread test.txt
Isi dari file test.txt adalah :
Nama = Afifah Ghaisani Imana, NIM = L200190198,
user@PC:~$
```

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

main(int argc, char const *argv[])
{
    int fd, i;
    char buf[100];
    if (argc < 2)
    {
        printf("Usage: ./a.out <filename>\n");
        exit(-1);
    }

    fd = open(argv[1], O_RDONLY);
    if (fd == -1)
    {
        printf("%s file does not exist\n", argv[1]);
        exit(-1);
    }

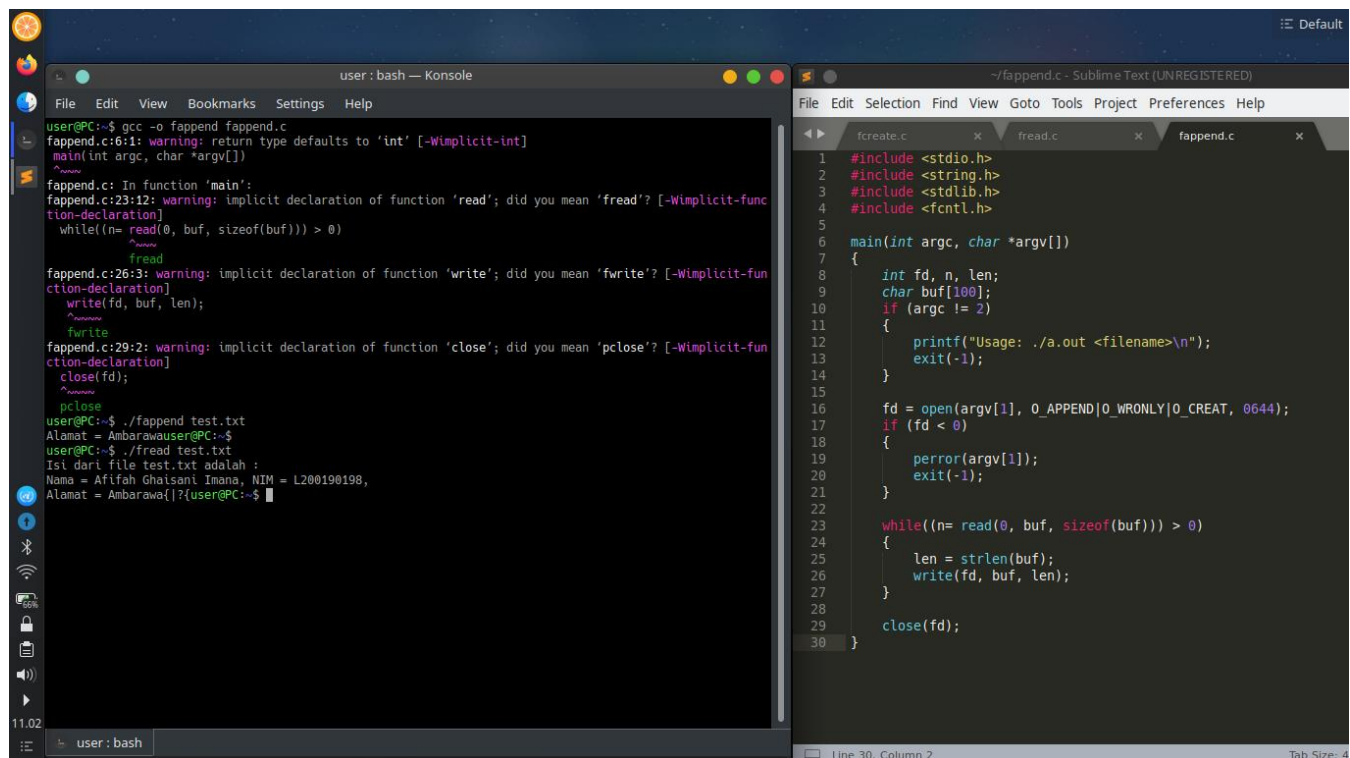
    printf("Isi dari file %s adalah : \n", argv[1]);
    while(read(fd, buf, sizeof(buf)) > 0)
        printf("%s", buf);

    close(fd);
}
```

3. Menambah isi file.

- ✓ Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.
- ✓ Gunakan nama-file sesuai dengan argumen yang diberikan dalam perintah command-line.
- ✓ Buka file di atas dengan menggunakan perintah open dengan opsi O_APPEND.
- ✓ Periksa file-descriptor, jika nilainya berupa angka negatif, stop program
- ✓ Baca input user dari console sampai user menekan tombol Ctrl + D. Jika user sudah menekan tombol Ctrl + D, baca 100 byte dari console dan simpan ke dalam variabel 'buf' menggunakan perintah read. Selanjutnya tuliskan isi variabel 'buf' ke dalam file menggunakan perintah 'write'.
- ✓ Tutup file dengan menggunakan fungsi 'close'.

Kode Program fappend.c dan tampilan saat di run pada terminal.



The image shows a terminal window on the left and a code editor on the right. The terminal window displays the compilation and execution of a C program named fappend.c. The code editor shows the source code of fappend.c, which includes headers for stdio.h, string.h, stdlib.h, andfcntl.h. The main function opens a file named test.txt in append mode, reads input from the user until Ctrl+D is pressed, writes the input to the file, and then closes the file.

```
user@PC:~$ gcc -o fappend fappend.c
fappend.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char *argv[])
^~~~~~
fappend.c: In function 'main':
fappend.c:23:12: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
while((n= read(0, buf, sizeof(buf))) > 0)
           ^~~~~~
fappend.c:26:3: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
write(fd, buf, len);
   ^~~~~~
fappend.c:29:2: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
close(fd);
   ^~~~~~
pclose
user@PC:~$ ./fappend test.txt
Alamat = Ambarawuser@PC:~$
user@PC:~$ ./fread test.txt
Isi dari file test.txt adalah :
Nama = Afifah Ghaisani Inana, NIM = L200190198,
Alamat = Ambarawa{ }?{user@PC:~$
```

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  #include <fcntl.h>
5
6  main(int argc, char *argv[])
7  {
8      int fd, n, len;
9      char buf[100];
10     if (argc != 2)
11     {
12         printf("Usage: ./a.out <filename>\n");
13         exit(-1);
14     }
15
16     fd = open(argv[1], O_APPEND|O_WRONLY|O_CREAT, 0644);
17     if (fd < 0)
18     {
19         perror(argv[1]);
20         exit(-1);
21     }
22
23     while((n= read(0, buf, sizeof(buf))) > 0)
24     {
25         len = strlen(buf);
26         write(fd, buf, len);
27     }
28
29     close(fd);
30 }
```