

**LAPORAN PRAKTIKUM
SISTEM OPERASI
MODUL 9
FILE SYSTEM CALL**



Disusun Oleh :

YESY LELY YESTIANA

L200210227

Kelas E

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2022/2023**

Lembar Kerja Praktikum Modul 9

NIM	: L200210227	Nilai praktek	:
Nama	: Yesy Lely Yestiana		
Dosen Pengampu	: Heru Setiya N., ST, M.Kom	Tanda tangan	:
Nama Asisten	: -		
Tanggal Praktikum	: 14/12/2022		

Tugas!!!

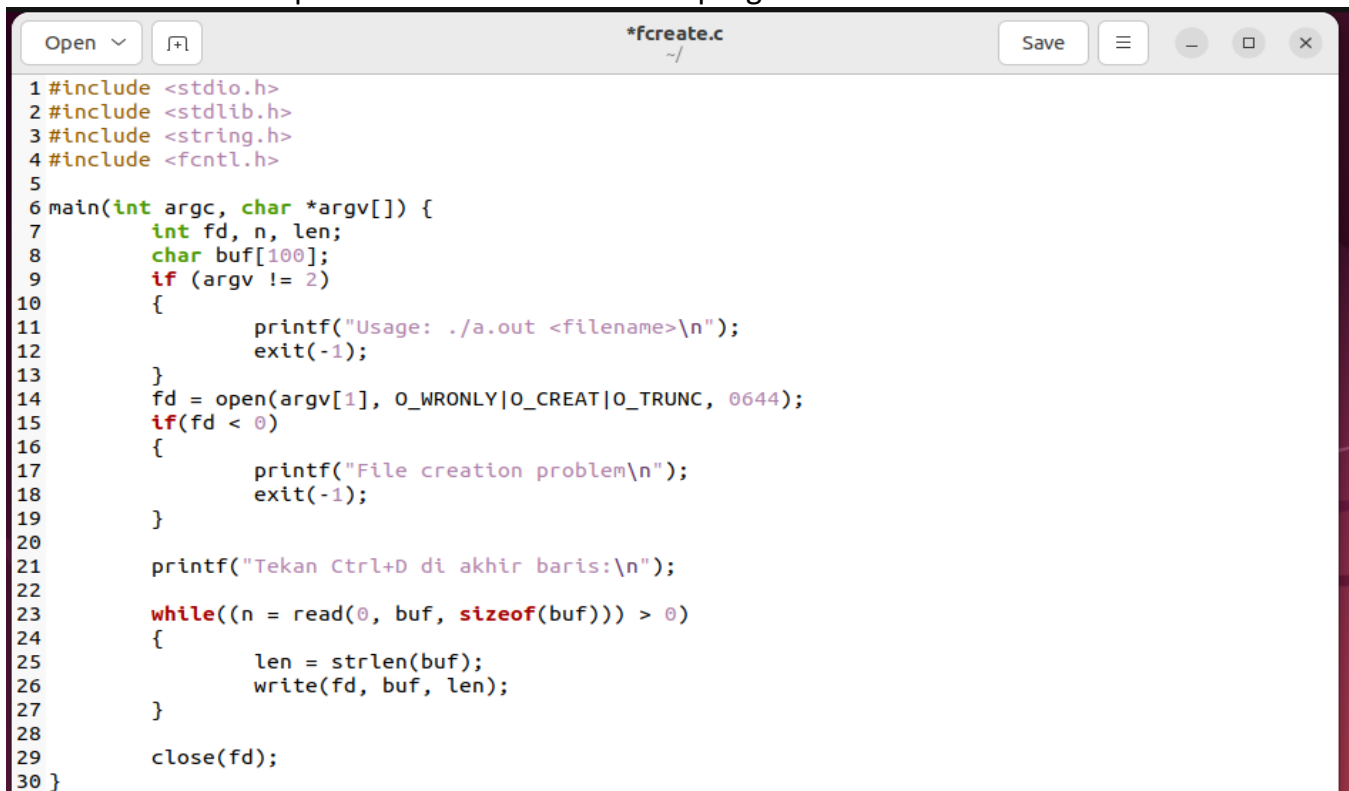
✓ **Membuat sebuah file dan menuliskan data didalamnya.**

Membuat program dengan algoritma sebagai berikut.

(contoh program diberikan pada bagian berikutnya) :

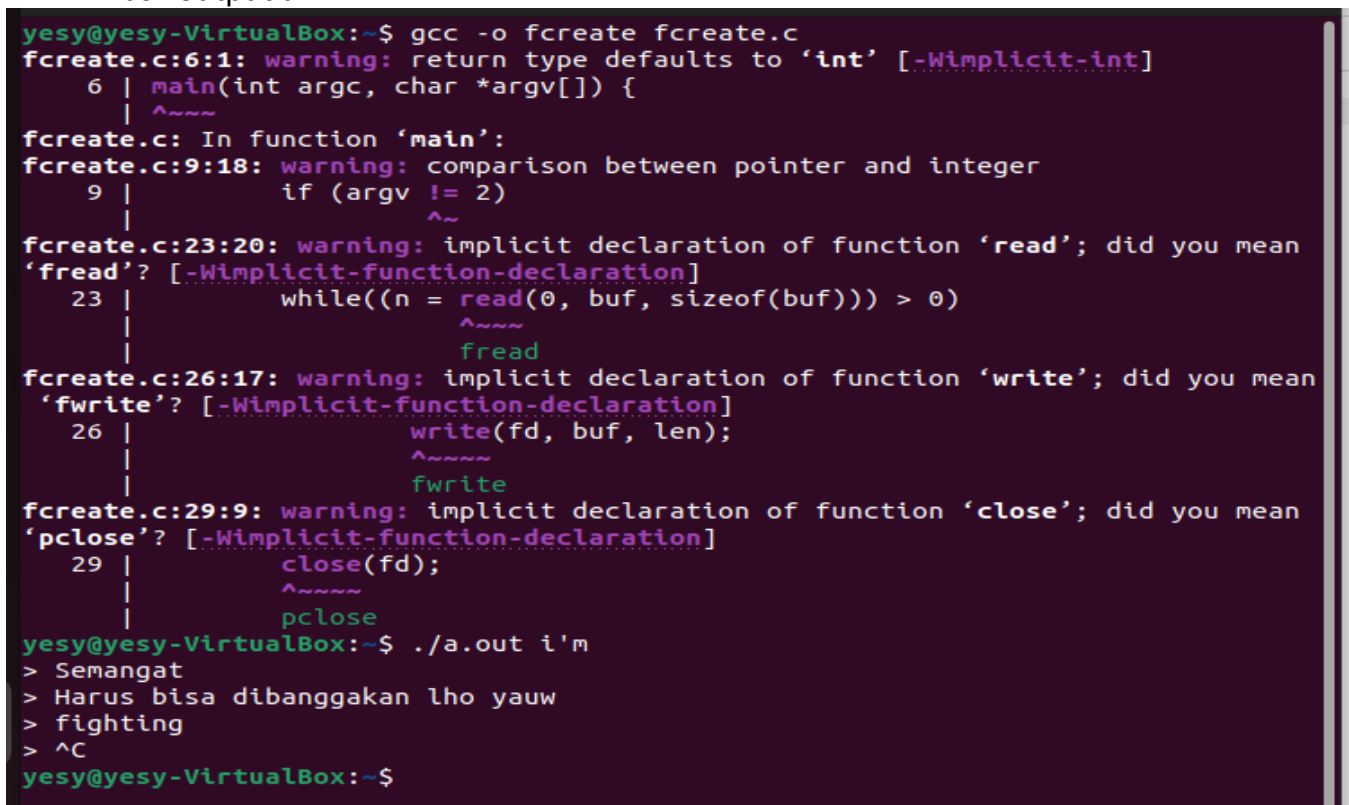
1. Deklarasi variabel untuk buffer 'buf' untuk menyimpan 100 byte data.
2. Gunakan nama file yang diberikan sebagai argumen pada command-line.
3. Buat sebuah file baru dengan nama file seperti pada (2) menggunakan fungsi system-call open() dengan opsi O_CREAT dan O_TRUNC.
4. Periksa 'file-descriptor', jika pembuatan file tidak berhasil, selanjutnya berhenti (stop) dan keluar dari program.
5. 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' dengan menggunakan perintah system-call read(). Memindahkan isi variabel 'buf' ke dalam file menggunakan perintah 'write'.
6. Menutup file dengan menggunakan fungsi 'close'.
7. Stop.

❖ Berikut merupakan screen shot dari kode program :



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

❖ Hasil output an :



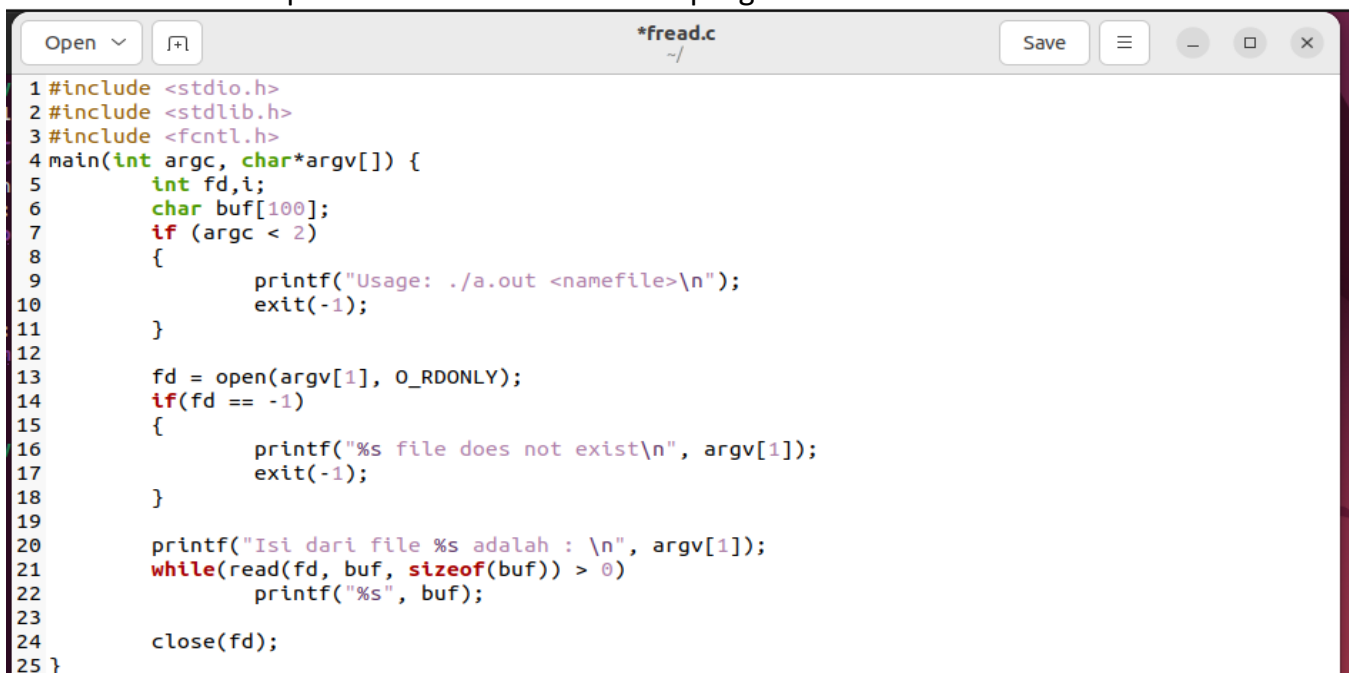
```
yesy@yesy-VirtualBox:~$ gcc -o fcreate fcreate.c
fcreate.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
  6 | main(int argc, char *argv[]) {
    | ^~~~~
fcreate.c: In function 'main':
fcreate.c:9:18: warning: comparison between pointer and integer
   9 |         if (argc != 2)
     |             ^~
fcreate.c:23:20: warning: implicit declaration of function 'read'; did you mean
'fread'? [-Wimplicit-function-declaration]
  23 |         while((n = read(0, buf, sizeof(buf))) > 0)
     |                   ^~~~~
     |                   fread
fcreate.c:26:17: warning: implicit declaration of function 'write'; did you mean
'fwrite'? [-Wimplicit-function-declaration]
  26 |         write(fd, buf, len);
     |         ^~~~~
     |         fwrite
fcreate.c:29:9: warning: implicit declaration of function 'close'; did you mean
'pclose'? [-Wimplicit-function-declaration]
  29 |         close(fd);
     |         ^~~~~
     |         pclose
yesy@yesy-VirtualBox:~$ ./a.out i'm
> Semangat
> Harus bisa dibanggakan lho yaww
> fighting
> ^C
yesy@yesy-VirtualBox:~$
```

✓ **Membaca sebuah file dan menampilkan isinya di layar.**

Membuat kode program dengan algoritma sebagai berikut :

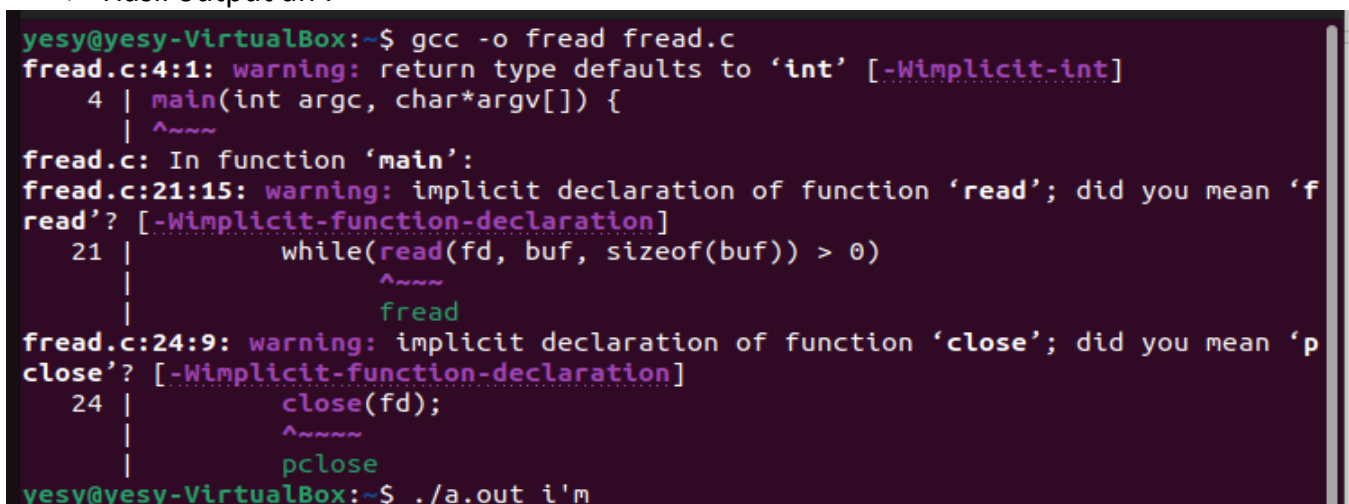
1. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.
2. Gunakan nama-file sesuai argumen yang diberikan dalam perintah command-line.
3. Buka file untuk dibaca menggunakan perintah 'open' dengan opsi O_RDONLY.
4. Periksa isi file-descriptor, Jika file tidak ada maka program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

❖ Berikut merupakan screen shot dari kode program :



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <fcntl.h>
4 main(int argc, char*argv[]) {
5     int fd,i;
6     char buf[100];
7     if (argc < 2)
8     {
9         printf("Usage: ./a.out <namefile>\n");
10        exit(-1);
11    }
12
13    fd = open(argv[1], O_RDONLY);
14    if(fd == -1)
15    {
16        printf("%s file does not exist\n", argv[1]);
17        exit(-1);
18    }
19
20    printf("Isi dari file %s adalah : \n", argv[1]);
21    while(read(fd, buf, sizeof(buf)) > 0)
22        printf("%s", buf);
23
24    close(fd);
25 }
```

❖ Hasil output an :



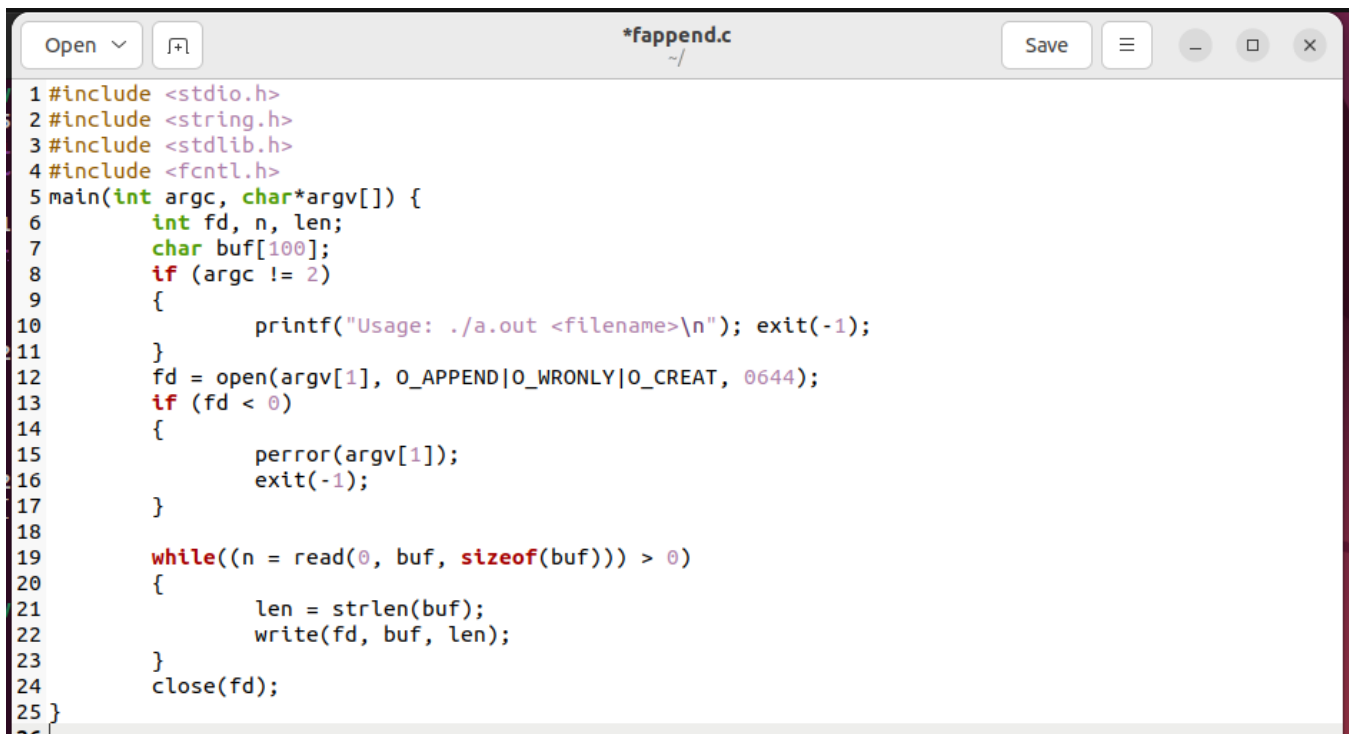
```
yesy@yesy-VirtualBox:~$ gcc -o fread fread.c
fread.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
  4 | main(int argc, char*argv[]) {
    | ^~~~~
fread.c: In function 'main':
fread.c:21:15: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
  21 |         while(read(fd, buf, sizeof(buf)) > 0)
    |               ^~~~~
    |               fread
fread.c:24:9: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
  24 |         close(fd);
    |         ^~~~~~
    |         pclose
yesy@yesy-VirtualBox:~$ ./a.out i'm
```

✓ Menambah isi file

Membuat kode program dengan algoritma sebagai berikut :

1. Deklarasi sebuah variabel buffer tipe character untuk menyimpan 100 byte data.
2. Gunakan nama-file sesuai argumen yang diberikan dalam perintah command-line.
3. Buka file di atas dengan menggunakan perintah open dengan opsi O_APPEND.
4. Periksa file-descriptor, jika nilainya berupa angka negatif, stop program.
5. 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'.
6. Tutup file dengan menggunakan fungsi 'close'.

❖ Berikut merupakan screen shot dari kode program :



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

❖ Hasil output an:

```
yesy@yesy-VirtualBox:~$ gcc -o fappend fappend.c
fappend.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
  5 | main(int argc, char*argv[]) {
    | ^~~~~
fappend.c: In function 'main':
fappend.c:19:20: warning: implicit declaration of function 'read'; did you mean
'fread'? [-Wimplicit-function-declaration]
  19 |         while((n = read(0, buf, sizeof(buf))) > 0)
      |                   ^~~~~
                        fread
fappend.c:22:17: warning: implicit declaration of function 'write'; did you mean
'fwrite'? [-Wimplicit-function-declaration]
  22 |         write(fd, buf, len);
      |         ^~~~~
                        fwrite
fappend.c:24:9: warning: implicit declaration of function 'close'; did you mean
'pclose'? [-Wimplicit-function-declaration]
  24 |         close(fd);
      |         ^~~~~
                        pclose
yesy@yesy-VirtualBox:~$ ./a.out i'm
```

✓ **Kesimpulan :**

File System Call merupakan fungsi system-call yang berkaitan dengan proses pembuatan, pembacaan dan penulisan file.