

LAPORAN PRAKTIKUM
MATA KULIAH PRAKTIKUM SISTEM OPERASI
MODUL 8 SYSTEM CALL



Disusun Oleh :
AFIFAH NUR NABILA
L200210249
Kelas E

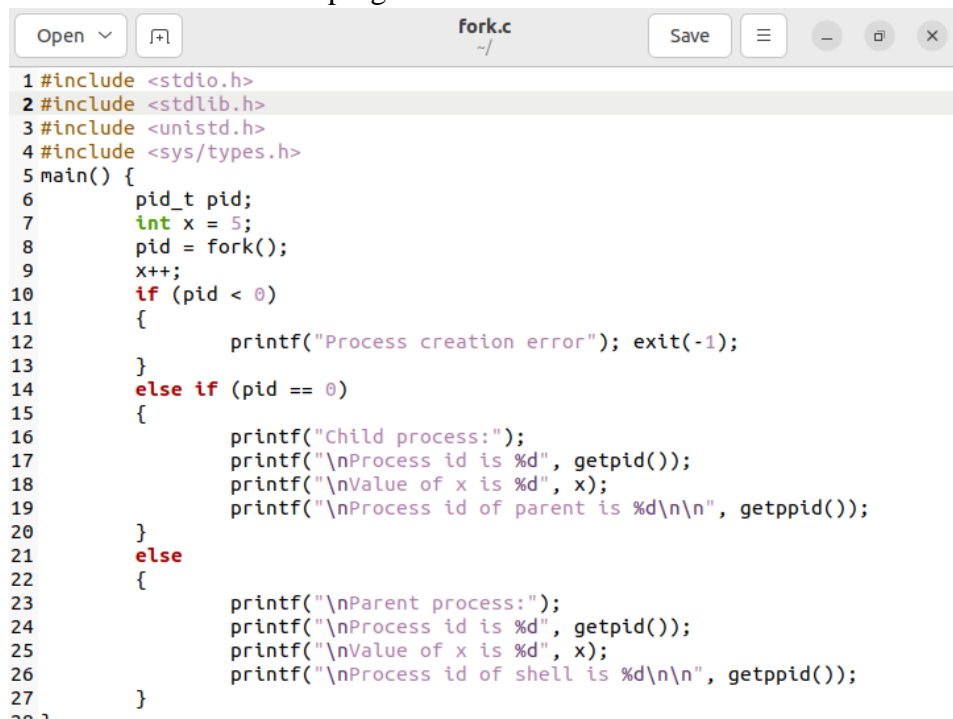
PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2022/2023

Laporan Praktikum Modul 8

NIM : L200210249 Nama : Afifah Nur Nabila Dosen Pengampu : Heru Setiya Nugraha, S.T, M.Kom Tanggal Praktikum : 6 Desember 2022	Nilai praktek : Tanda tangan :
---	---

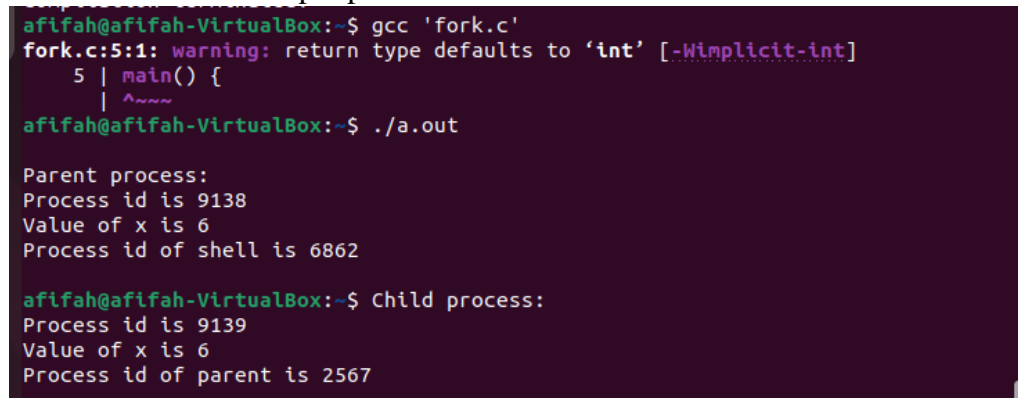
Langkah Kerja

1. Membuat sebuah 'child process' (proses baru) dengan menggunakan system call 'fork'.
⇒ Berikut *screenshot* kode program dalam teks editor



```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 main() {
6     pid_t pid;
7     int x = 5;
8     pid = fork();
9     x++;
10    if (pid < 0)
11    {
12        printf("Process creation error"); exit(-1);
13    }
14    else if (pid == 0)
15    {
16        printf("Child process:");
17        printf("\nProcess id is %d", getpid());
18        printf("\nValue of x is %d", x);
19        printf("\nProcess id of parent is %d\n\n", getppid());
20    }
21    else
22    {
23        printf("\nParent process:");
24        printf("\nProcess id is %d", getpid());
25        printf("\nValue of x is %d", x);
26        printf("\nProcess id of shell is %d\n\n", getppid());
27    }
28 }
```

- ⇒ Berikut *screenshot* output pada terminal



```
afifah@afifah-VirtualBox:~$ gcc 'fork.c'
fork.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
    5 | main() {
      | ~~~~
afifah@afifah-VirtualBox:~$ ./a.out

Parent process:
Process id is 9138
Value of x is 6
Process id of shell is 6862

afifah@afifah-VirtualBox:~$ Child process:
Process id is 9139
Value of x is 6
Process id of parent is 2567
```

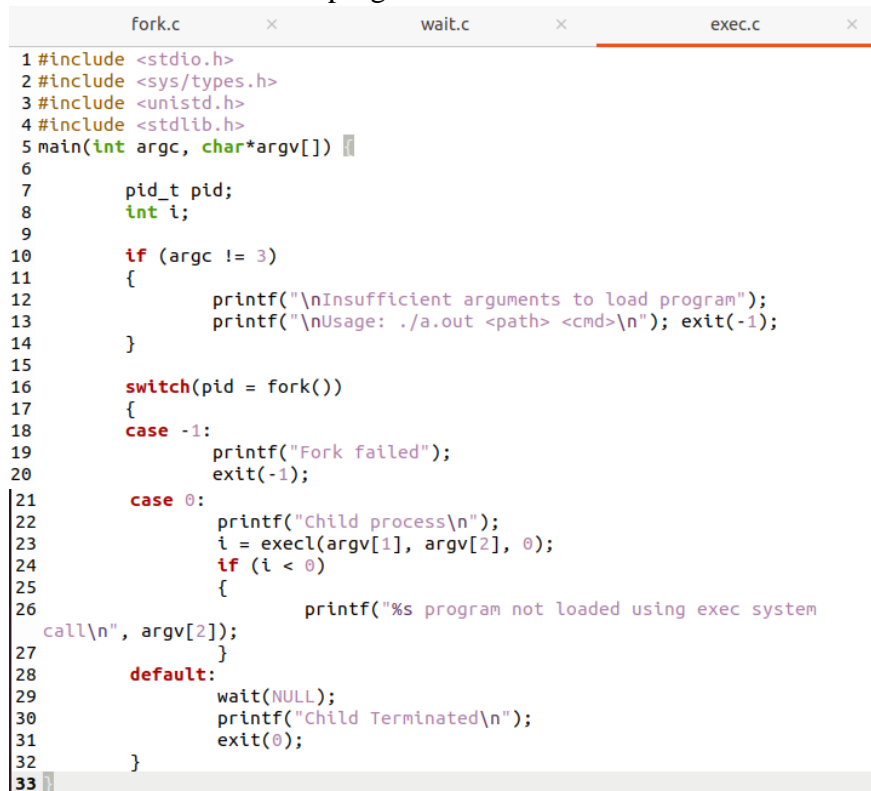
2. Menghentikan sementara (block) proses parent sampai dengan proses child selesai, menggunakan perintah system call 'wait'.

⇒ Berikut *screenshot* kode program dalam teks editor

⇒ Berikut *screenshot* output pada terminal

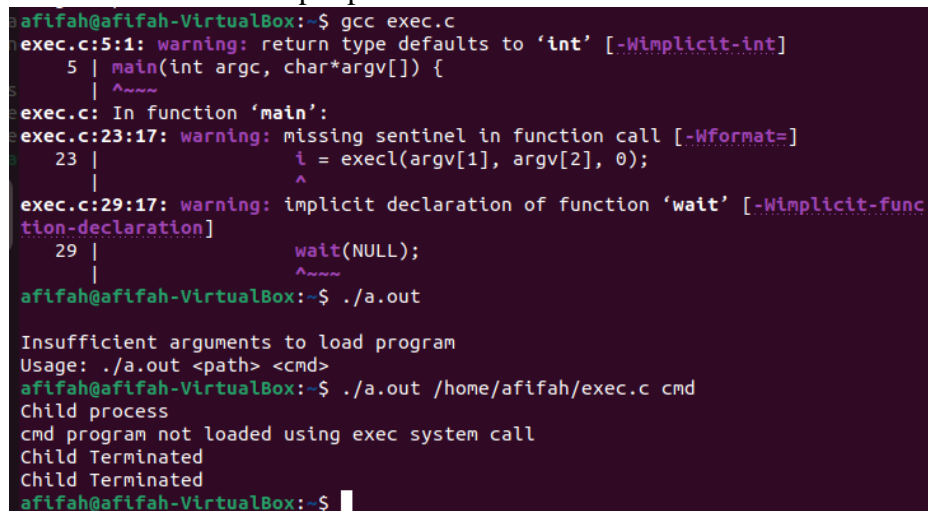
3. Loading program yang dapat dieksekusi dalam sebuah 'child' proses menggunakan perintah system call 'exec'.

⇒ Berikut *screenshot* kode program dalam teks editor



```
1 #include <stdio.h>
2 #include <sys/types.h>
3 #include <unistd.h>
4 #include <stdlib.h>
5 main(int argc, char*argv[]) {
6
7     pid_t pid;
8     int i;
9
10    if (argc != 3)
11    {
12        printf("\nInsufficient arguments to load program");
13        printf("\nUsage: ./a.out <path> <cmd>\n"); exit(-1);
14    }
15
16    switch(pid = fork())
17    {
18        case -1:
19            printf("Fork failed");
20            exit(-1);
21
22        case 0:
23            printf("Child process\n");
24            i = execl(argv[1], argv[2], 0);
25            if (i < 0)
26            {
27                printf("%s program not loaded using exec system
28                call\n", argv[2]);
29            }
30
31        default:
32            wait(NULL);
33            printf("Child Terminated\n");
34            exit(0);
35    }
```

⇒ Berikut *screenshot* output pada terminal



```
afifah@afifah-VirtualBox:~$ gcc exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
    5 | main(int argc, char*argv[]) {
      | ^~~~~
exec.c: In function 'main':
exec.c:23:17: warning: missing sentinel in function call [-Wformat=]
    23 |         i = execl(argv[1], argv[2], 0);
      |         ^
exec.c:29:17: warning: implicit declaration of function 'wait' [-Wimplicit-func
tion-declaration]
    29 |         wait(NULL);
      |         ^~~~~
afifah@afifah-VirtualBox:~$ ./a.out

Insufficient arguments to load program
Usage: ./a.out <path> <cmd>
afifah@afifah-VirtualBox:~$ ./a.out /home/afifah/exec.c cmd
Child process
cmd program not loaded using exec system call
Child Terminated
Child Terminated
afifah@afifah-VirtualBox:~$
```

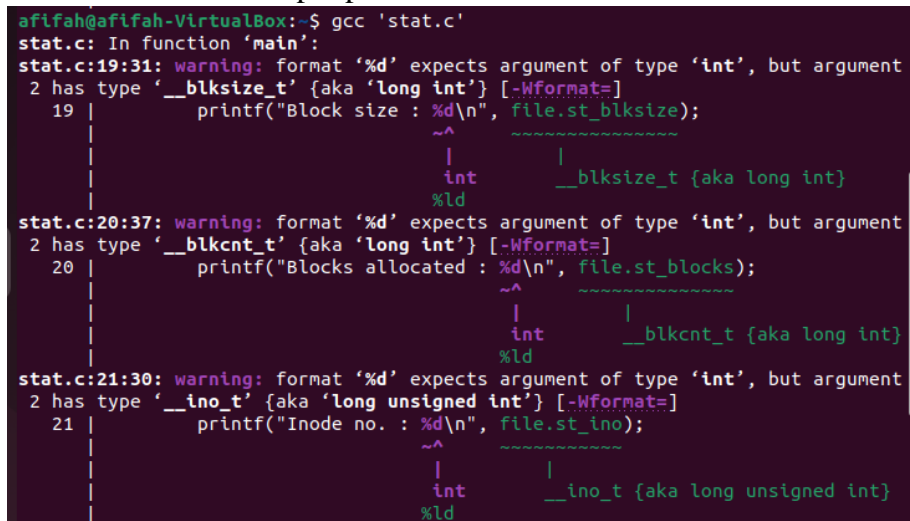
4. Menampilkan status file menggunakan perintah system call 'stat'.

⇒ Berikut *screenshot* kode program dalam teks editor



```
1 #include <stdio.h>
2 #include <sys/stat.h>
3 #include <stdlib.h>
4 #include <time.h>
5 int main(int argc, char*argv[]) {
6     struct stat
7     file; int n;
8     if (argc != 2)
9     {
10         printf("Usage: ./a.out <filename>\n"); exit(-1);
11     }
12     if ((n = stat(argv[1], &file)) == -1)
13     {
14         perror(argv[1]);
15         exit(-1);
16     }
17     printf("User id : %d\n", file.st_uid);
18     printf("Group id : %d\n", file.st_gid);
19     printf("Block size : %d\n", file.st_blksize);
20     printf("Blocks allocated : %d\n", file.st_blocks);
21     printf("Inode no. : %d\n", file.st_ino);
22     printf("Last accessed : %s", ctime(&(file.st_atime)));
23     printf("Last modified : %s", ctime(&(file.st_mtime)));
24     printf("File size : %d bytes\n", file.st_size);
25     printf("No. of links : %d\n", file.st_nlink);
26     printf("Permissions : ");
27     printf( (S_ISDIR(file.st_mode)) ? "d" : "-");
28     printf( (file.st_mode & S_IRUSR) ? "r" : "-");
29     printf( (file.st_mode & S_IWUSR) ? "w" : "-");
30     printf( (file.st_mode & S_IXUSR) ? "x" : "-");
31     printf( (file.st_mode & S_IRGRP) ? "r" : "-");
32     printf( (file.st_mode & S_IWGRP) ? "w" : "-");
33     printf( (file.st_mode & S_IXGRP) ? "x" : "-");
34     printf( (file.st_mode & S_IROTH) ? "r" : "-");
35     printf( (file.st_mode & S_IWOTH) ? "w" : "-");
36     printf( (file.st_mode & S_IXOTH) ? "x" : "-");
37     printf("\n");
38     if(file.st_mode & S_IFREG)
39         printf("File type : Regular\n");
40     if(file.st_mode & S_IFDIR)
41         printf("File type : Directory\n");
42 }
```

⇒ Berikut *screenshot* output pada terminal



```
afifah@afifah-VirtualBox:~$ gcc 'stat.c'
stat.c: In function 'main':
stat.c:19:31: warning: format '%d' expects argument of type 'int', but argument
 2 has type '__blksize_t' {aka 'long int'} [-Wformat=]
19 |         printf("Block size : %d\n", file.st_blksize);
   |                               ~^
   |                               |
   |                               int      __blksize_t {aka long int}
   |                               %ld
stat.c:20:37: warning: format '%d' expects argument of type 'int', but argument
 2 has type '__blkcnt_t' {aka 'long int'} [-Wformat=]
20 |         printf("Blocks allocated : %d\n", file.st_blocks);
   |                               ~^
   |                               |
   |                               int      __blkcnt_t {aka long int}
   |                               %ld
stat.c:21:30: warning: format '%d' expects argument of type 'int', but argument
 2 has type '__ino_t' {aka 'long unsigned int'} [-Wformat=]
21 |         printf("Inode no. : %d\n", file.st_ino);
   |                               ~^
   |                               |
   |                               int      __ino_t {aka long unsigned int}
   |                               %ld
```

```

stat.c:24:30: warning: format '%d' expects argument of type 'int', but argument
2 has type '__off_t' {aka 'long int'} [-Wformat=]
24 |     printf("File size : %d bytes\n", file.st_size);
    |                                ^~
    |                                |
    |                                int
    |                                |
    |                                __off_t {aka long int}
    |                                %ld
stat.c:25:33: warning: format '%d' expects argument of type 'int', but argument
2 has type '__nlink_t' {aka 'long unsigned int'} [-Wformat=]
25 |     printf("No. of links : %d\n", file.st_nlink);
    |                                ^~
    |                                |
    |                                int
    |                                |
    |                                __nlink_t {aka long unsigned
int}
    |                                %ld
afifah@afifah-VirtualBox:~$ ./a.out
Usage: ./a.out <filename>
afifah@afifah-VirtualBox:~$ ./a.out stat.c
User id : 1000
Group id : 1000
Block size : 4096
Blocks allocated : 8
Inode no. : 414412
Last accessed : Tue Dec  6 10:03:57 2022
Last modified : Tue Dec  6 10:03:48 2022
File size : 1370 bytes
No. of links : 1
Permissions : -rw-rw-r--
File type : Regular

```

5. Menampilkan isi direktori menggunakan perintah system call 'readdir'.
 ⇒ Berikut *screenshot* kode program dalam teks editor

```

fork.c x    wait.c x    exec.c x    stat.c x    dirlist.c x
1 #include <stdio.h>
2 #include <dirent.h>
3 #include <stdlib.h>
4 main(int argc, char*argv[]) {
5     struct dirent *dptr;
6     DIR *dname;
7
8     if (argc != 2)
9     {
10         printf("Usage: ./a.out <dirname>\n");
11         exit(-1);
12     }
13     if ((dname = opendir (argv[1])) == NULL)
14     {
15         perror(argv[1]);
16         exit(-1);
17     }
18     while(dptr=readdir(dname))
19         printf("%s\n", dptr->d_name);
20
21     closedir(dname);
22 }

```

- ⇒ Berikut *screenshot* output pada terminal

```

afifah@afifah-VirtualBox:~$ gcc dirlist.c
dirlist.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
4 | main(int argc, char*argv[]) {
  | ^~~~~
afifah@afifah-VirtualBox:~$ ./a.out /home/afifah/downloads
/home/afifah/downloads: No such file or directory
afifah@afifah-VirtualBox:~$ ./a.out /home
..
.
afifah
12002102491
afifah@afifah-VirtualBox:~$ ./a.out /home/afifah/Pictures
..
Screenshots
.

```