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

Open | fork.c ~/ | Save

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
main() {
        pid_t pid;
        int x = 5;
        pid = fork();
        x++;
        if (pid < 0)
        {
                printf("Process creation error"); exit(-1);
        }
        else if (pid == 0)
        {
                printf("Child process:");
                printf("\nProcess id is %d", getpid());
                printf("\nValue of x is %d", x);
                printf("\nProcess id of parent is %d\n\n", getppid());
        }
        else
        {
                printf("\nParent process:");
                printf("\nProcess id is %d", getpid());
                printf("\nValue of x is %d", x);
                printf("\nProcess id of shell is %d\n\n", getppid());
        }
}
```

   ⇨ 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. Menghantikan sementara (block) proses parent sampai dengan proses child selesai, menggunakan perintah system call ‘wait’.
   ⇨ Berikut *screenshot* kode program dalam teks editor

Open | wait.c ~/ | Save

fork.c | wait.c

```
1
2 #include <stdio.h>
3 #include <stdlib.h>
4 #include <unistd.h>
5 #include <sys/types.h>
6 #include <sys/wait.h>
7 main() {
8         int i, status;
9         pid_t pid;
10        pid = fork();
11
12        if (pid < 0) {
13                printf("\nPembuatan proses gagal\n");
14                exit(-1);
15        }
16        else if(pid > 0)
17        {
18                wait(NULL);
19                printf("\nParent starts\nNomor Genap:");
20                for (i=2;i<=10;i+=2)
21                        printf ("%3d", i);
22                printf("\nParent ends\n");
23        }
24        else if(pid == 0)
25        {
26                printf("\nChild starts\nNomor Ganjil:");
27                for (i=1;i<=10;i+=2)
28                        printf ("%3d", i);
29                printf("\nChild ends\n");
30        }
31 }
```

   ⇨ Berikut *screenshot* output pada terminal

```
afifah@afifah-VirtualBox:~$ gcc 'wait.c'
wait.c:7:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
    7 | main() {
      | ^~~~
afifah@afifah-VirtualBox:~$ ./a.out

Child starts
Nomor Ganjil:  1  3  5  7  9
Child ends

Parent starts
Nomor Genap:  2  4  6  8 10
Parent ends
afifah@afifah-VirtualBox:~$
```

3. Loading program yang dapat dieksekusi dalam sebuah ‘child’ proses menggunakan perintah system call ‘exec’.
   - ⇨ Berikut *screenshot* kode program dalam teks editor

fork.c | wait.c | exec.c

```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
main(int argc, char*argv[]) {

        pid_t pid;
        int i;

        if (argc != 3)
        {
                printf("\nInsufficient arguments to load program");
                printf("\nUsage: ./a.out <path> <cmd>\n"); exit(-1);
        }

        switch(pid = fork())
        {
        case -1:
                printf("Fork failed");
                exit(-1);
        case 0:
                printf("Child process\n");
                i = execl(argv[1], argv[2], 0);
                if (i < 0)
                {
                        printf("%s program not loaded using exec system call\n", argv[2]);
                }
        default:
                wait(NULL);
                printf("Child Terminated\n");
                exit(0);
        }
}
```

   - ⇨ 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-function-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

fork.c × wait.c × exec.c × stat.c ×

```c
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <time.h>
int main(int argc, char*argv[]) {
        struct stat
        file; int n;
        if (argc != 2)
        {
                printf("Usage: ./a.out <filename>\n"); exit(-1);
        }
        if ((n = stat(argv[1], &file)) == -1)
        {
                perror(argv[1]);
                exit(-1);
        }
        printf("User id : %d\n", file.st_uid);
        printf("Group id : %d\n", file.st_gid);
        printf("Block size : %d\n", file.st_blksize);
        printf("Blocks allocated : %d\n", file.st_blocks);
        printf("Inode no. : %d\n", file.st_ino);
        printf("Last accessed : %s", ctime(&(file.st_atime)));
        printf("Last modified : %s", ctime(&(file.st_mtime)));
        printf("File size : %d bytes\n", file.st_size);
        printf("No. of links : %d\n", file.st_nlink);
        printf("Permissions : ");
        printf( (S_ISDIR(file.st_mode)) ? "d" : "-");
        printf( (file.st_mode & S_IRUSR) ? "r" : "-");
        printf( (file.st_mode & S_IWUSR) ? "w" : "-");
        printf( (file.st_mode & S_IXUSR) ? "x" : "-");
        printf( (file.st_mode & S_IRGRP) ? "r" : "-");
        printf( (file.st_mode & S_IWGRP) ? "w" : "-");
        printf( (file.st_mode & S_IXGRP) ? "x" : "-");
        printf( (file.st_mode & S_IROTH) ? "r" : "-");
        printf( (file.st_mode & S_IWOTH) ? "w" : "-");
        printf( (file.st_mode & S_IXOTH) ? "x" : "-");
        printf("\n");
        if(file.st_mode & S_IFREG)
                printf("File type : Regular\n");
        if(file.st_mode & S_IFDIR)
                printf("File type : Directory\n");
}
```

   ⇨ 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 | wait.c | exec.c | stat.c | dirlist.c

```c
#include <stdio.h>
#include <dirent.h>
#include <stdlib.h>
main(int argc, char*argv[]) {
        struct dirent *dptr;
        DIR *dname;

        if (argc != 2)
        {
                printf("Usage: ./a.out <dirname>\n");
                exit(-1);
        }
        if ((dname = opendir (argv[1])) == NULL)
        {
                perror(argv[1]);
                exit(-1);
        }
        while(dptr=readdir(dname))
                printf("%s\n", dptr->d_name);

        closedir(dname);
}
```

   - ⇨ 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
l2002102491
afifah@afifah-VirtualBox:~$ ./a.out /home/afifah/Pictures
..
Screenshots
.
```