

LAPORAN PRAKTIKUM SISTEM OPERASI MODUL 8

UCOK SAYUTI

L200210239

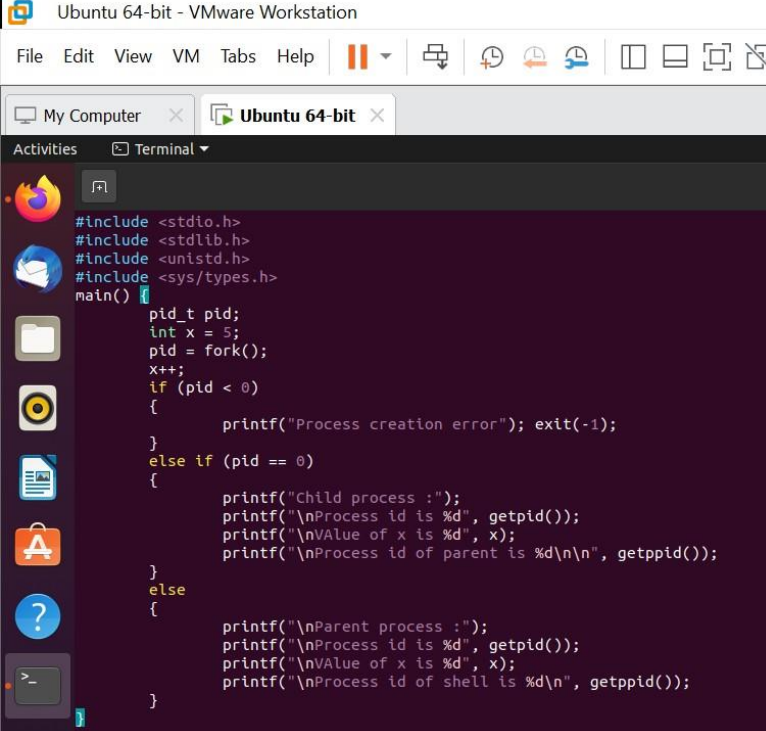
KELAS E

LANGKAH KERJA

1. Membuat sebuah 'child process' (proses baru) dengan menggunakan system call 'fork'.

Membuat program dengan algoritma sebagai berikut : (contoh program diberikan pada bagaian berikutnya).

- Deklarasi sebuah variable x yang akan diakses Bersama antara child procese dan parent process.
- Membuat sebuah child process menggunakan system call fork.
- Jika return value bernlai -1, tampilkan teks 'Pembuatan process GAGAL', dilanjutkan dengan keluar program dengan perintah system call 'exit'.
- Jka return value sama dengan 0 (NOL), tampilkan teks 'child process' ,tampilkan ID process dari child process menggunakan perintah system call 'getpid', tampilkan nilai x, dan tampilkan ID process parent dengan perintah system call 'getpid'.
- Untuk nilai return value lainnya, tampilkan teks 'parent process', tampilkan ID dari parent process menggunakan perintah system call getpid, tempilkan nila x, dan tampilkan ID dari process shell menguunakan perintah system call 'getpid'.
- Stop.



```
#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", 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", getpid());
    }
}
```

```

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

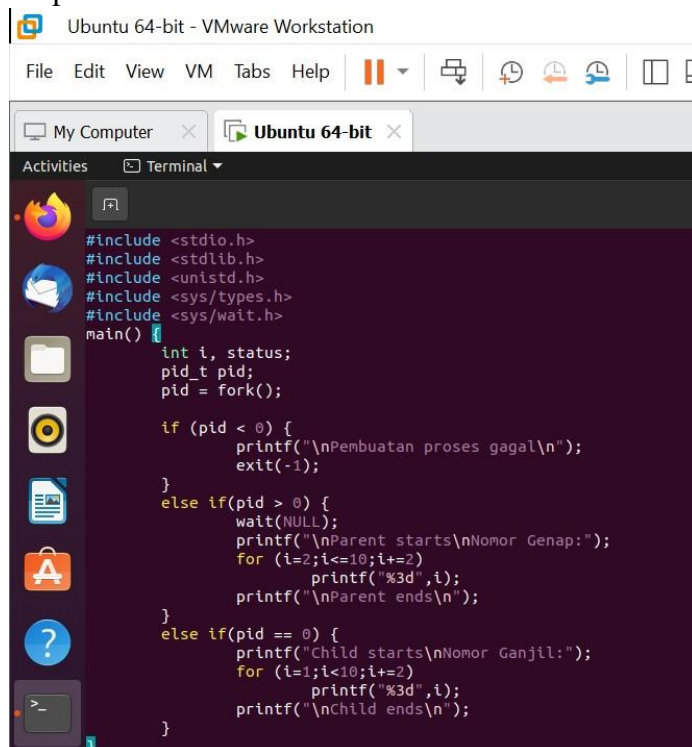
Parent process:
Process id is 7965
Value of x is 6
Process id of shell is 7918
Child process:
Process id is 7966
Value of x is 6
Process id of parent is 1295
ubuntu@ubuntu:~$

```

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

Membuat program dengan algoritma sebagai berikut : (contoh program diberikan pada bagian berikutnya).

- Membuat sebuah chld process menggunakan system call 'fork'.
- Jika return value bernilai -1, selanjutnya tampilkan teks 'pembuatan proses gagal', dan keluar program dengan menggunakan perintah system call 'exit'.
- Jika return berupa angka positif (>0), 'pause' hentikan sementara 'parent' proses tunggu sampai child process berakhir dengan menggunakan perintah system call 'wait'. Tampilkan teks 'Parent starts', selanjutnya tampilkan nomor genap mulai dari 0 – 10, terakhir tampilkan teks 'Parent end'.
- Jka return value bernilai 0 (NOL), tampilkan teks "Child start", tampilkan nomor ganjil mulai dari 0 – 10, selanjutnya tampilkan teks 'child ends'.
- Stop.



```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
main() {
    int i, status;
    pid_t pid;
    pid = fork();

    if (pid < 0) {
        printf("\nPembuatan proses gagal\n");
        exit(-1);
    }
    else if (pid > 0) {
        wait(NULL);
        printf("\nParent starts\nNomor Genap:");
        for (i=2; i<=10; i+=2)
            printf("%3d", i);
        printf("\nParent ends\n");
    }
    else if (pid == 0) {
        printf("\nChild starts\nNomor Ganjil:");
        for (i=1; i<=10; i+=2)
            printf("%3d", i);
        printf("\nChild ends\n");
    }
}

```

```

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

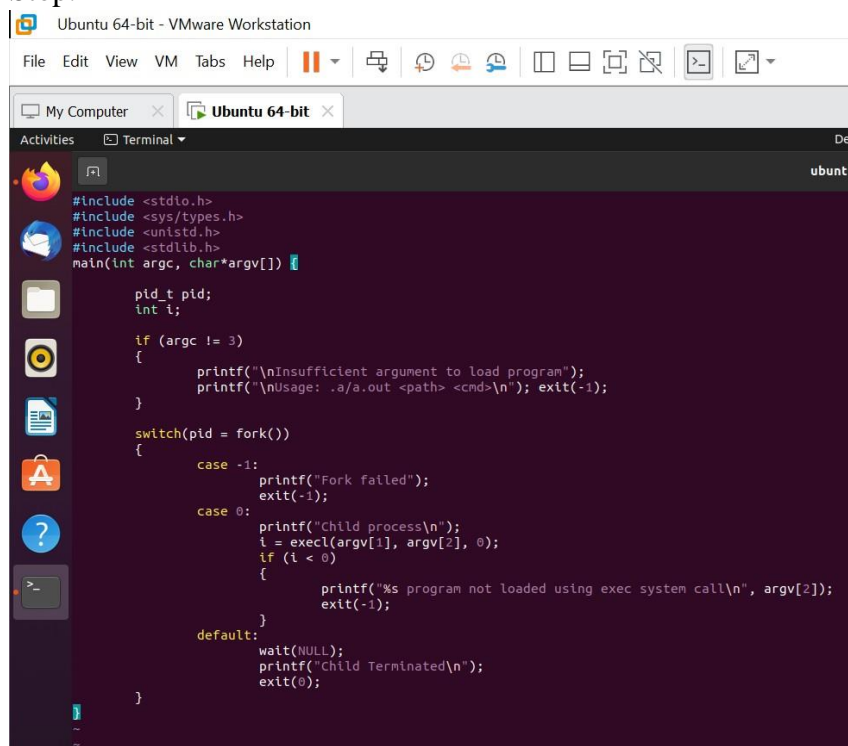
Parent process:
Process id is 7965
Value of x is 6
Process id of shell is 7918
Child process:
Process id is 7966
Value of x is 6
Process id of parent is 1295
ubuntu@ubuntu:~$

```

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

Membuat program dengan algoritma sebagai berikut : (contoh program diberikan pada bagian berikutnya).

- Jika terdapat 3 argumen dalam command-line berhenti (Stop).
- Membuat child process dengan perintah system call 'fork'.
- Jika return value adalah -1, selanjutnya tampilkan teks 'Pembuatan proses Gagal', dan keluar program dengan perintah system call 'fork'.
- Jika return value > 0 (positif), selanjutnya hentikan parent-process sementara hingga child-process berakhir dengan menggunakan perintah system call 'wait'. Tampilkan teks 'Child berakhir', dan hentikan parent-process.
- Jika return value sama dengan 0 (NOL), selanjutnya tampilkan teks 'Child starts', load program dari lokasi yang diberikan dalam 'path' ke dalam child-process, menggunakan perintah system call 'exec'. Jika return value dari perintah 'exec' adalah bilangan negative, tampilkan error yang terjadi dan stop. Hentikan child-process.
- Stop.



```

Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
My Computer x Ubuntu 64-bit x
Activities Terminal
#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 argument 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]);
                exit(-1);
            }
        default:
            wait(NULL);
            printf("Child Terminated\n");
            exit(0);
    }
}

```

```

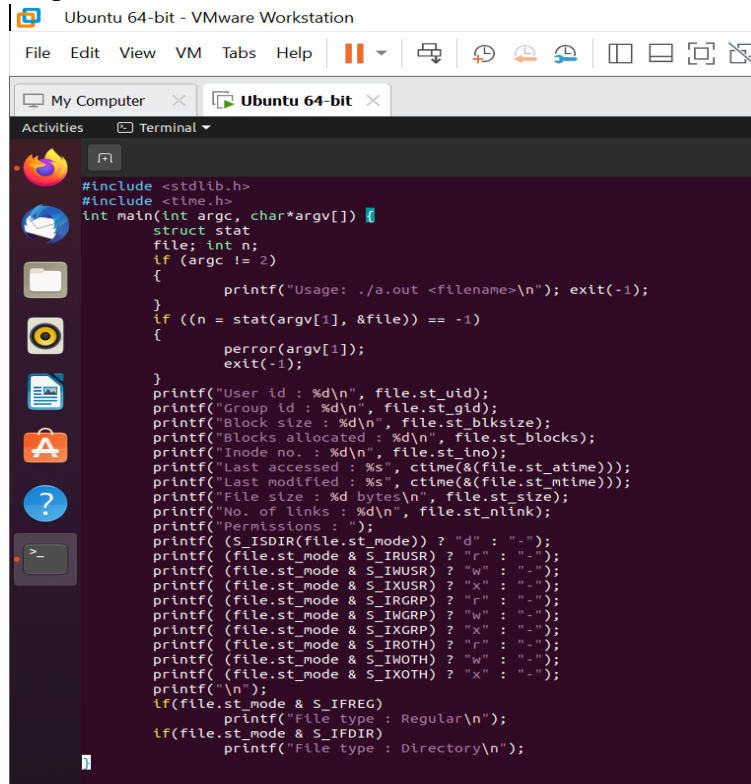
ubuntu@ubuntu:~$ vim exec.c
ubuntu@ubuntu:~$ gcc exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
  5 | main(int argc, char*argv[]) {
    | ^~~~~
exec.c: In function 'main':
exec.c:23:4: warning: missing sentinel in function call [-Wformat=]
  23 |     i = execl(argv[1], argv[2], 0);
    |           ^
exec.c:30:4: warning: implicit declaration of function 'wait' [-Wimplicit-function-declaration]
  30 |     wait(NULL);
    |     ^~~~~
ubuntu@ubuntu:~$ ./a.out /bin/ls ls
Child process
a.out Desktop Documents Downloads exec.c fork.c Music Pictures Public snap Templates Videos wait.c
Child Terminated
ubuntu@ubuntu:~$

```

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

Membuat program dengan algoritma sebagai berikut: (contoh code ada di bagian berikutnya).

- Gunakan 'nama file' yang diberikan melalui argument dalam perintah command-line.
- Jika 'nama-file' tidak ada maka stop disini (keluar program).
- Panggil system call 'stat' pada 'nama-file' tersebut yang akan mengembalikan sebuah struktur.
- Tampilkan informasi mengenai st_uid, st_blksize, st_block, st_size, st_nlink, etc.
- Ubah waktu dalam st_time, st_mtime dengan menggunakan fungsi ctime.
- Bandingkans t_mode dengan konstanta mode seperti S_IRUSR, S_IWGRP, S_IXOTH dan tampilkan informasi mengenai 'file-permissions'.
- Stop.



```

Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
My Computer x Ubuntu 64-bit x
Activities x Terminal x
#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");
}

```

```

ubuntu@ubuntu:~$ vim stat.c
ubuntu@ubuntu:~$ gcc stat.c
stat.c: In function 'main':
stat.c:19:24: 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}
stat.c:20:30: 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}
stat.c:21:23: 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}
stat.c:24:23: 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}
stat.c:25:26: 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}
ubuntu@ubuntu:~$ ./a.out stat.c
ubuntu@ubuntu:~$ ./a.out stat.c
User id : 1000
Group id : 1000
Block size : 4096
Blocks allocated : 8
Inode no. : 276045
Last accessed : Tue Dec  8 04:56:22 2020
Last modified : Tue Dec  8 04:55:42 2020
File size : 1371 bytes
No. of links : 1
Permissions : -rw-rw-r--
File type : Regular
ubuntu@ubuntu:~$

```

5. Menampilkan isi direktori menggunakan perintah system call 'readdir'.

Membuat program dengan algoritma sebagai berikut : (contoh code ada di bagian berikutnya).

- Gunakan 'nama-direktori' yang diberikan sebagai argument pada command-line.
- Jika direktori tidak ditemukan stop, keluar program
- Buka direktori menggunakan perintah system call 'opendir' yang akan menghasilkan sebuah struktur.
- Baca direktori menggunakan perintah system call 'readdir' yang juga akan menghasilkan struktur data.
- Tampilkan d_name (nama direktori)
- Akhiri pembaca direktori dengan perintah system call 'closedir'
- Stop.

Ubuntu 64-bit - VMware Workstation

File Edit View VM Tabs Help

My Computer Ubuntu 64-bit

Activities Terminal

```
#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);
}
```

```
ubuntu@ubuntu:~$ vim dirlist.c
ubuntu@ubuntu:~$ gcc dirlist.c
dirlist.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
 4 | main(int argc, char*argv[]) {
    | ^~~~~
ubuntu@ubuntu:~$ ./a.out Ahyati
ABOUT-NLS
NEWS
missing
Makefile.in
INSTALL
aclocal.m4
TODO
m4
THANKS
COPYING.DOC
configure
AUTHORS
install-sh
config.h.in
ChangeLog
.
config.guess
configure.ac
Makefile.am
config.rpath
README
nano-5.4
depcomp
po
src
..
lib
doc
syntax
IMPROVEMENTS
COPYING
config.sub
ubuntu@ubuntu:~$
```

