LAPORAN PRAKTIKUM SISTEM OPERASI

MODUL 8: SYSTEM CALL

Disusun oleh:

Nama : Riyan Catur Agusta

NIM : L200210262

Kelas : Praktikum SO E

PROGRAM STUDI TEKNIK INFORMATIKA

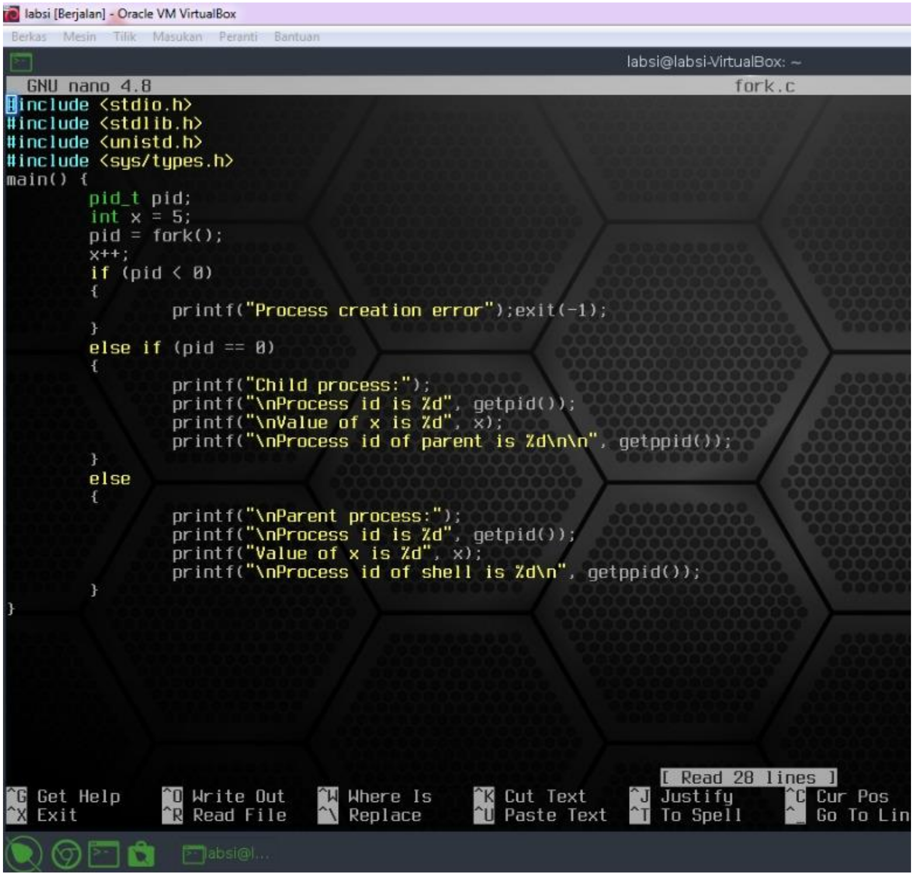
FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA TAHUN 2022/2023

LANGKAH KERJA

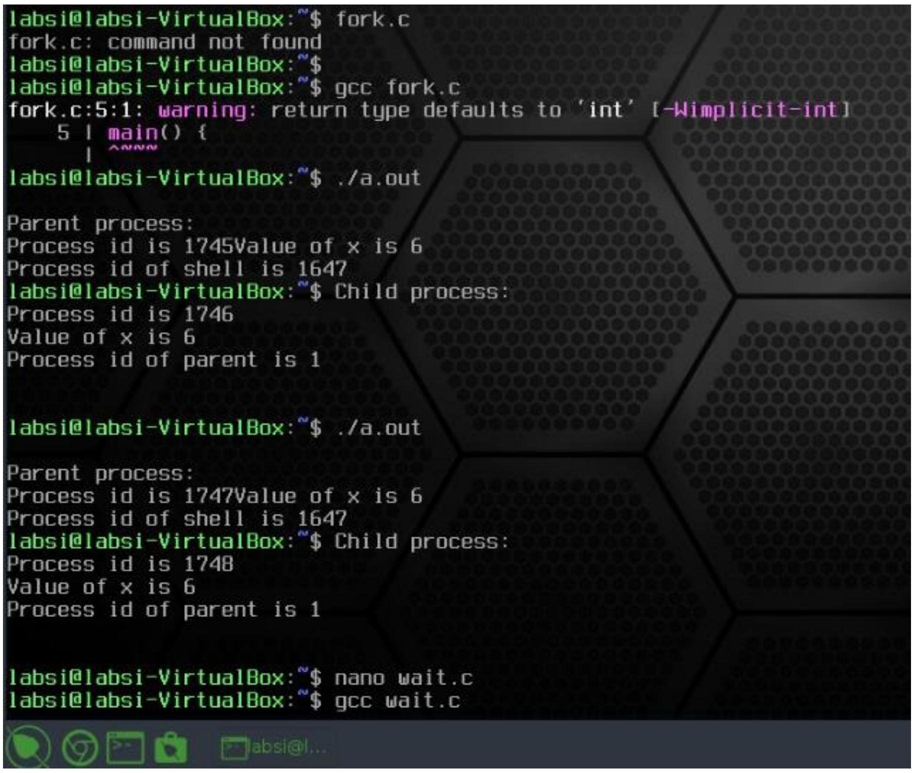
1. Membuat sebuah “Child Process” (Proses Baru) dengan menggunakan system call “fork”.

   Kode Program:



labsi [Berjalan] - Oracle VM VirtualBox
Berkas Mesin Tilik Masukan Peranti Bantuan
labsi@labsi-VirtualBox: ~
GNU nano 4.8
fork.c

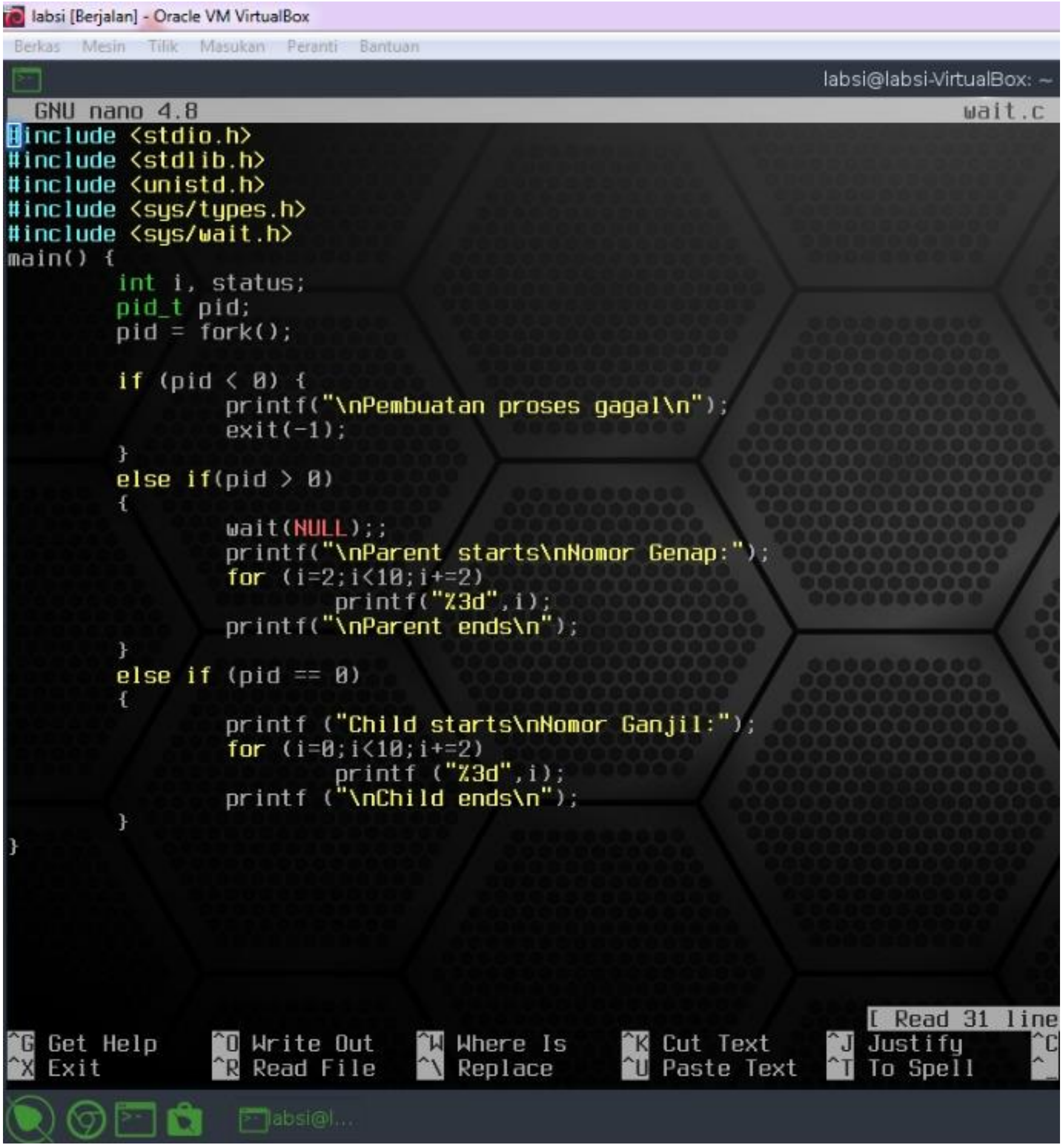
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
main() {
        pid_t pid;
        int x = 5;
        pid = fork();
        x++;
        if (pid < 0)
        {
                printf("Process creation error");exit(-1);
        }
        else if (pid == 0)
     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



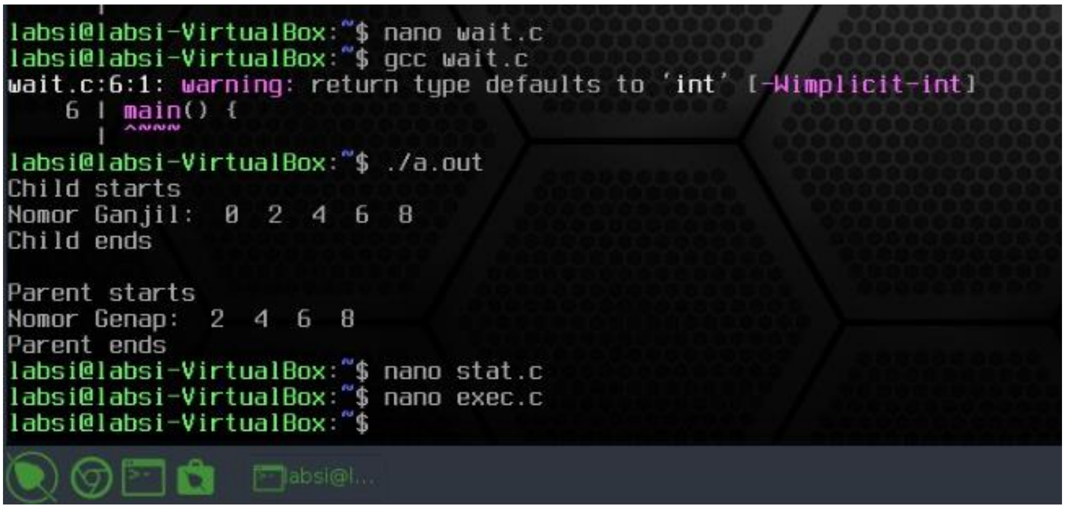
```
labsi@labsi-VirtualBox:~$ fork.c
fork.c: command not found
labsi@labsi-VirtualBox:~$
labsi@labsi-VirtualBox:~$ gcc fork.c
fork.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
    5 | main() {
      | ^~~~
labsi@labsi-VirtualBox:~$ ./a.out

Parent process:
Process id is 1745Value of x is 6
Process id of shell is 1647
labsi@labsi-VirtualBox:~$ Child process:
Process id is 1746
Value of x is 6
Process id of parent is 1


labsi@labsi-VirtualBox:~$ ./a.out

Parent process:
Process id is 1747Value of x is 6
Process id of shell is 1647
labsi@labsi-VirtualBox:~$ Child process:
Process id is 1748
Value of x is 6
Process id of parent is 1


labsi@labsi-VirtualBox:~$ nano wait.c
labsi@labsi-VirtualBox:~$ gcc wait.c
```

2. Menghentikan sementara (Block) proses parent sampai dengan proses child selesai, menggunakan perintah system call ‘wait’.

Kode Program:



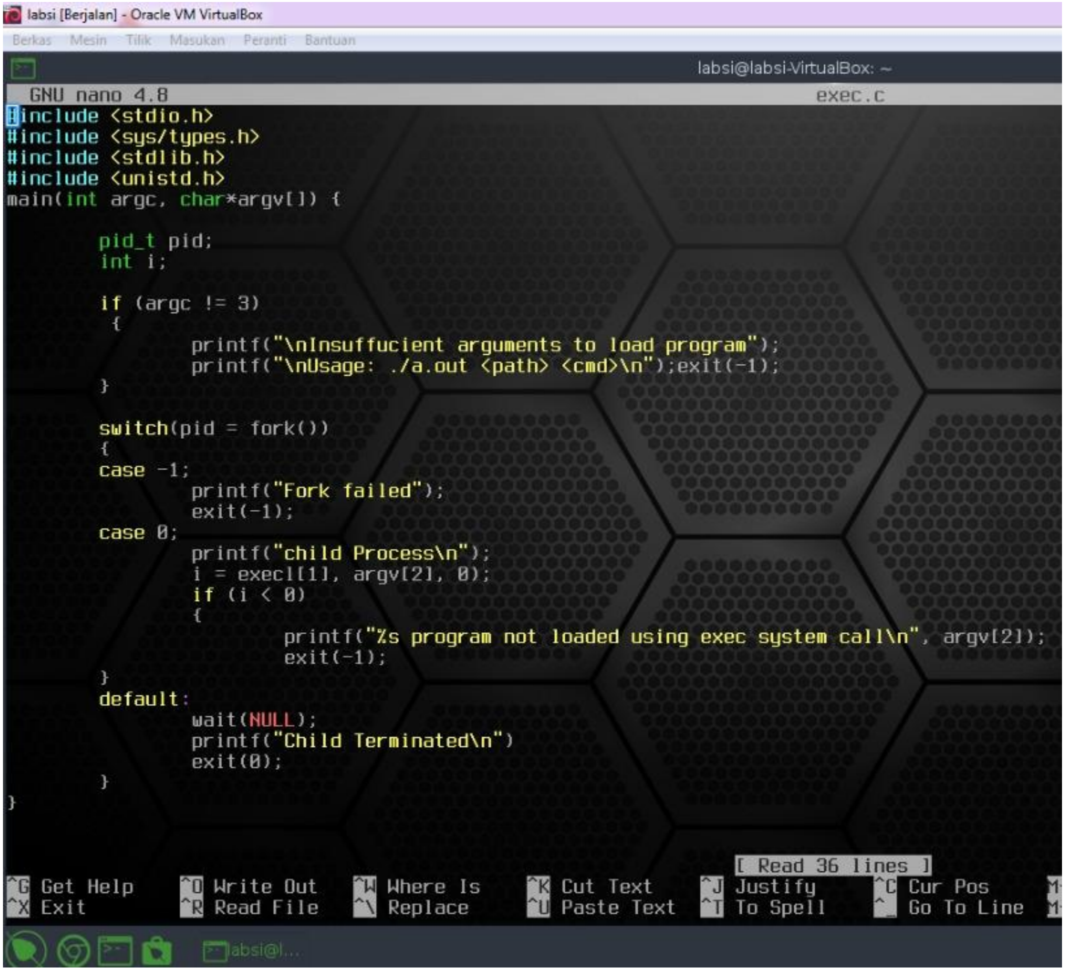
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
main() {
        int i, status;
        pid_t pid;
        pid = fork();

        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 ("Child starts\nNomor Ganjil:");
                for (i=0;i<10;i+=2)
                        printf ("%3d",i);
                printf ("\nChild ends\n");
        }
}
```



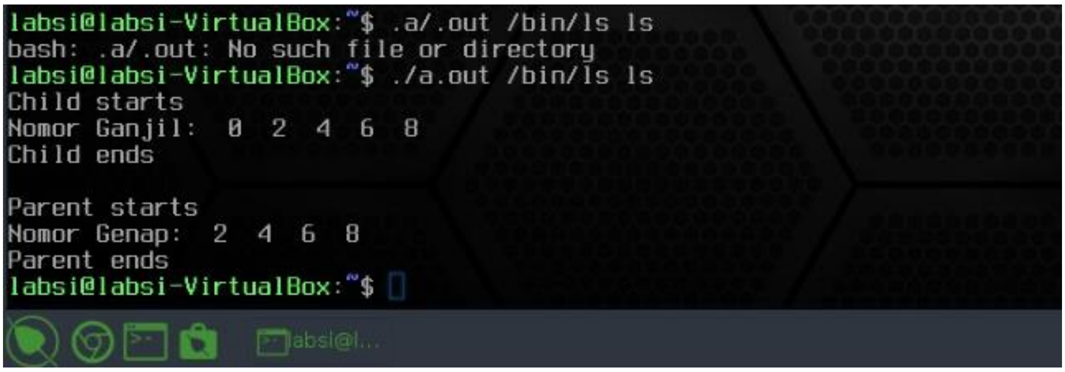
```
labsi@labsi-VirtualBox:~$ nano wait.c
labsi@labsi-VirtualBox:~$ gcc wait.c
wait.c:6:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
    6 | main() {
      | ^~~~
labsi@labsi-VirtualBox:~$ ./a.out
Child starts
Nomor Ganjil:  0  2  4  6  8
Child ends

Parent starts
Nomor Genap:  2  4  6  8
Parent ends
labsi@labsi-VirtualBox:~$ nano stat.c
labsi@labsi-VirtualBox:~$ nano exec.c
labsi@labsi-VirtualBox:~$
```

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

Kode Program:



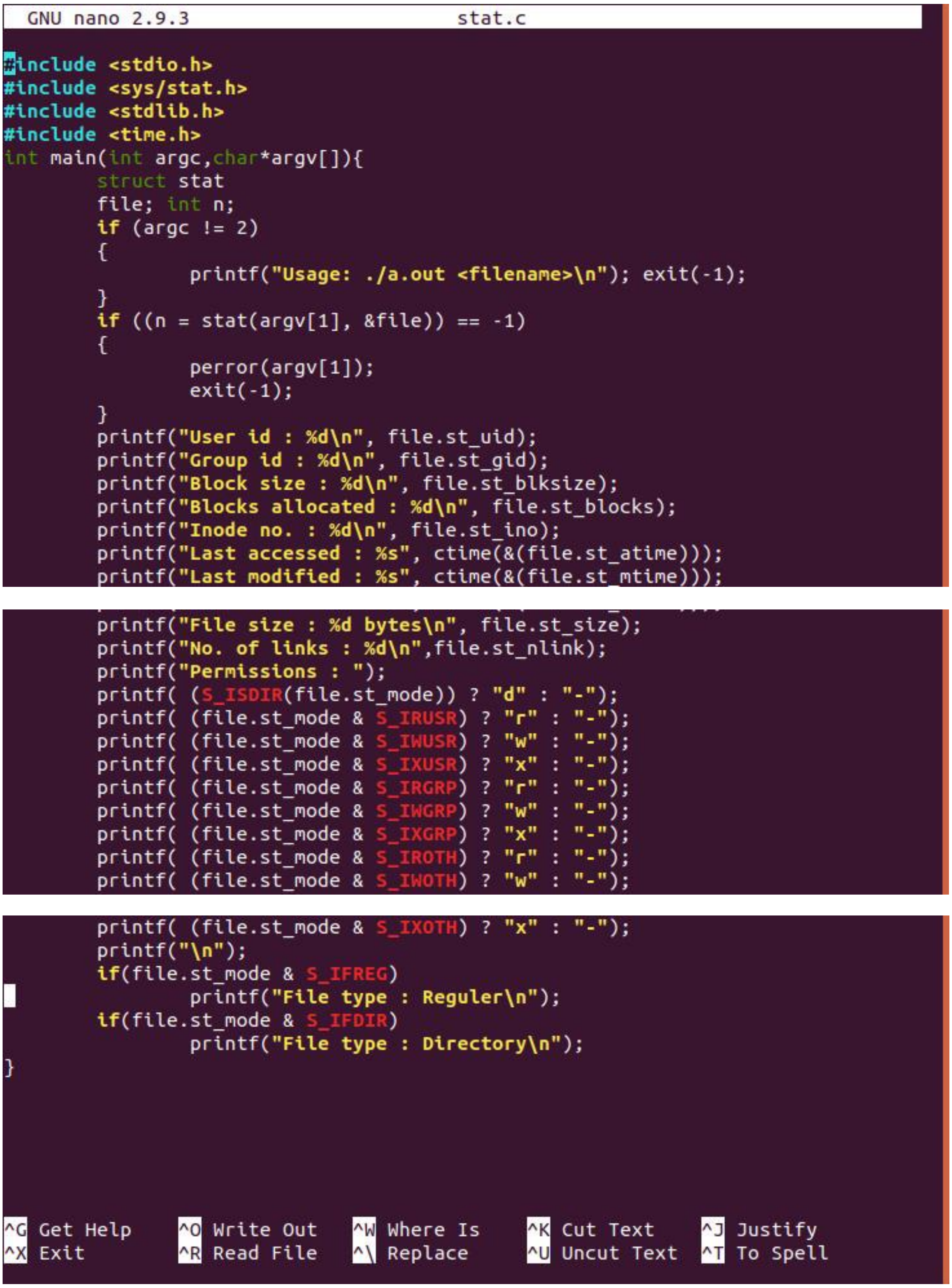
```
#include <stdio.h>
#include <sys/types.h>
#include <stdlib.h>
#include <unistd.h>
main(int argc, char*argv[]) {

        pid_t pid;
        int i;

        if (argc != 3)
         {
                printf("\nInsuffucient 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[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);
        }
}
```



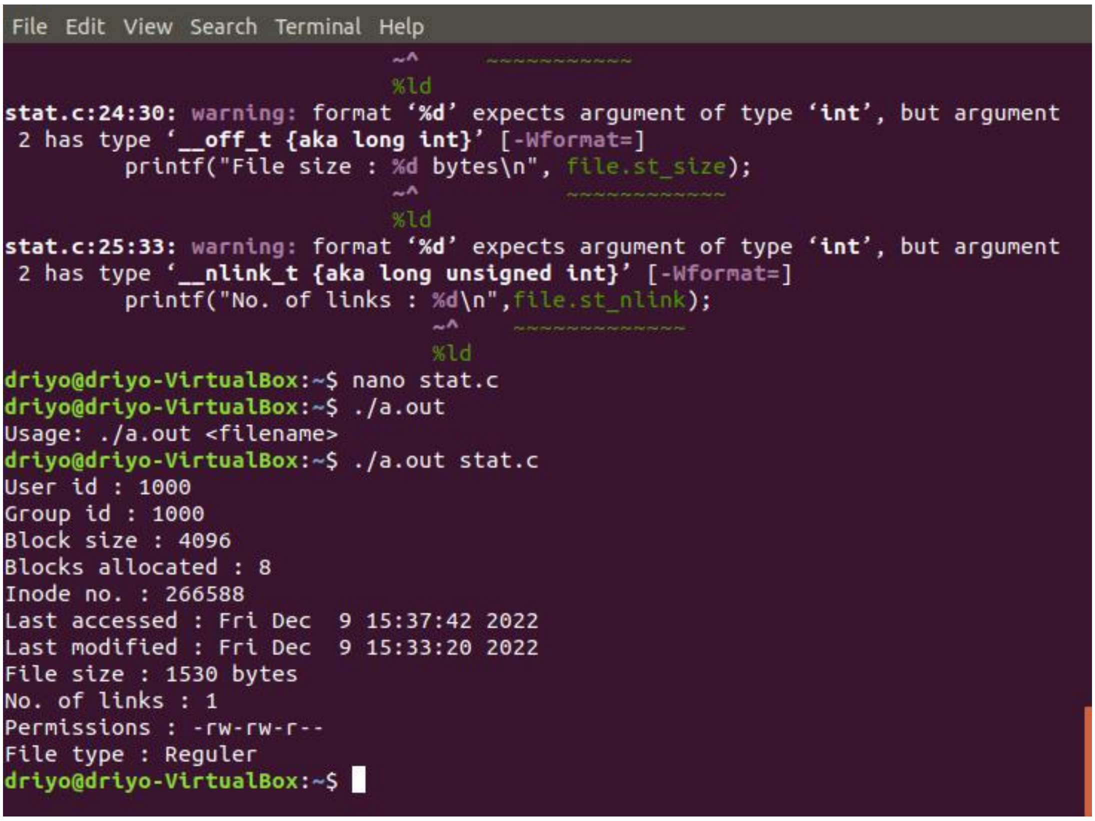
```
labsi@labsi-VirtualBox:~$ .a/.out /bin/ls ls
bash: .a/.out: No such file or directory
labsi@labsi-VirtualBox:~$ ./a.out /bin/ls ls
Child starts
Nomor Ganjil:  0  2  4  6  8
Child ends

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

4. Menampilkan status file menggunakan perintah system cell ‘stat’.

Kode Program:



```
#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 : Reguler\n");
        if(file.st_mode & S_IFDIR)
                printf("File type : Directory\n");
}
```

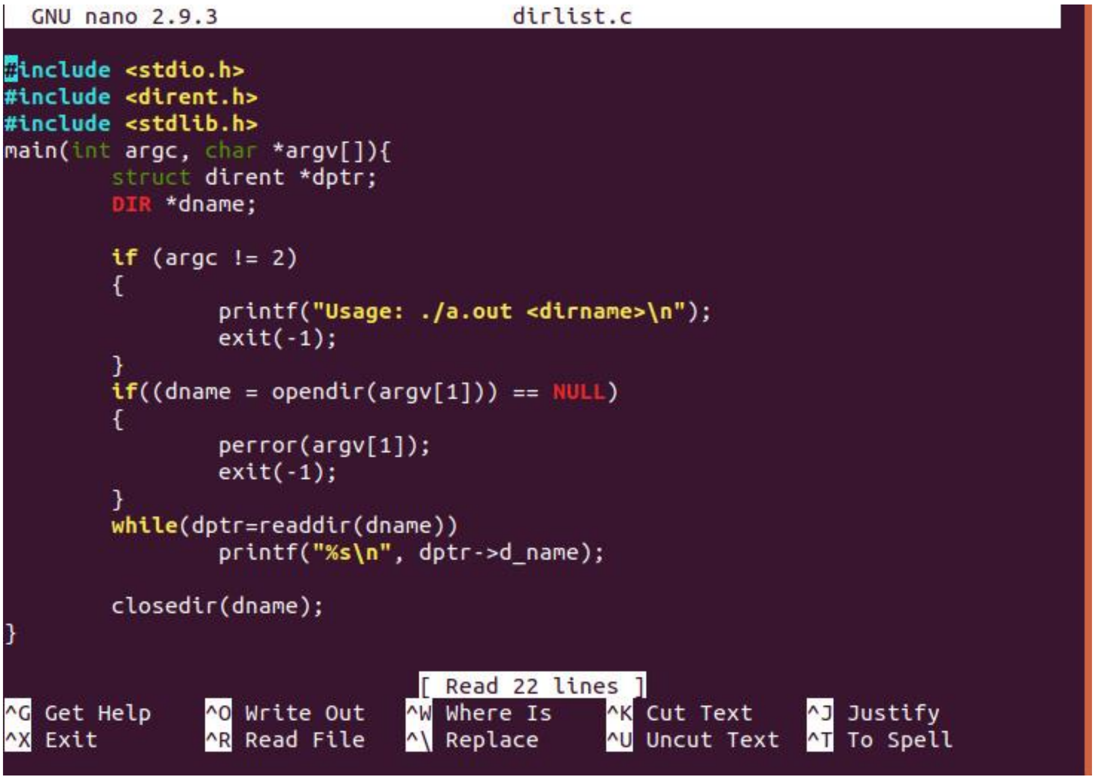


```
                              ~^      ~~~~~~~~~~~
                              %ld
stat.c:24:30: warning: format ‘%d’ expects argument of type ‘int’, but argument
 2 has type ‘__off_t {aka long int}’ [-Wformat=]
        printf("File size : %d bytes\n", file.st_size);
                              ~^            ~~~~~~~~~~~~
                              %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=]
        printf("No. of links : %d\n",file.st_nlink);
                                 ~^   ~~~~~~~~~~~~~
                                 %ld
driyo@driyo-VirtualBox:~$ nano stat.c
driyo@driyo-VirtualBox:~$ ./a.out
Usage: ./a.out <filename>
driyo@driyo-VirtualBox:~$ ./a.out stat.c
User id : 1000
Group id : 1000
Block size : 4096
Blocks allocated : 8
Inode no. : 266588
Last accessed : Fri Dec  9 15:37:42 2022
Last modified : Fri Dec  9 15:33:20 2022
File size : 1530 bytes
No. of links : 1
Permissions : -rw-rw-r--
File type : Reguler
driyo@driyo-VirtualBox:~$
```

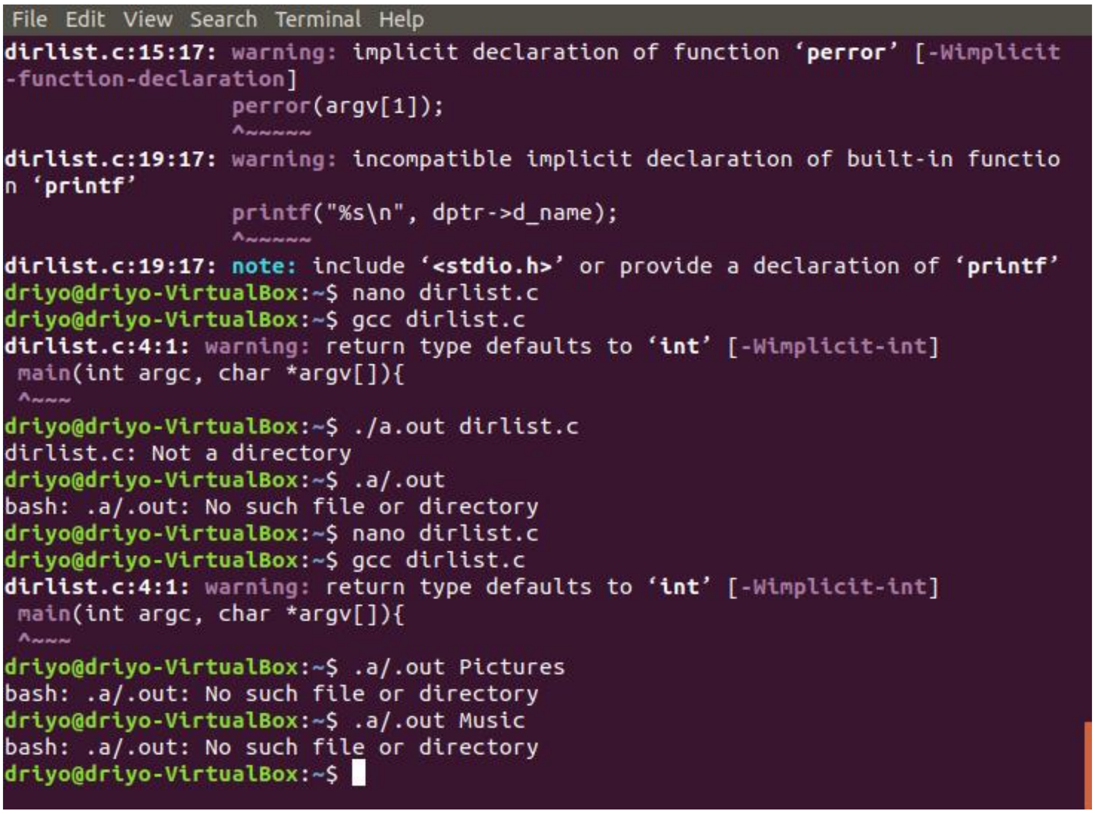
Note: Ini memakai laptop temen saya soalnya laptop saya kentang. Tapi ini saya mengerjakan sendiri mas.

5. Menampilkan isi direktori menggunakan perintah system call ‘readdir’.

Kode Program:



```
  GNU nano 2.9.3                      dirlist.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);
}

                              [ Read 22 lines ]
```



```
dirlist.c:15:17: warning: implicit declaration of function ‘perror’ [-Wimplicit
-function-declaration]
                 perror(argv[1]);
                 ^~~~~~
dirlist.c:19:17: warning: incompatible implicit declaration of built-in functio
n ‘printf’
                 printf("%s\n", dptr->d_name);
                 ^~~~~~
dirlist.c:19:17: note: include ‘<stdio.h>’ or provide a declaration of ‘printf’
driyo@driyo-VirtualBox:~$ nano dirlist.c
driyo@driyo-VirtualBox:~$ gcc dirlist.c
dirlist.c:4:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
 main(int argc, char *argv[]){
 ^~~~
driyo@driyo-VirtualBox:~$ ./a.out dirlist.c
dirlist.c: Not a directory
driyo@driyo-VirtualBox:~$ .a/.out
bash: .a/.out: No such file or directory
driyo@driyo-VirtualBox:~$ nano dirlist.c
driyo@driyo-VirtualBox:~$ gcc dirlist.c
dirlist.c:4:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
 main(int argc, char *argv[]){
 ^~~~
driyo@driyo-VirtualBox:~$ .a/.out Pictures
bash: .a/.out: No such file or directory
driyo@driyo-VirtualBox:~$ .a/.out Music
bash: .a/.out: No such file or directory
driyo@driyo-VirtualBox:~$ 
```

Note: Ini memakai laptop temen saya soalnya laptop saya kentang. Tapi ini saya mengerjakan sendiri mas.