PRAKTIKUM SISTEM OPERASI

MODUL 8

SYSTEM CALL

Oleh :

Dian Putri Mutiara Hapsari

L200210238

PROGRAM STUDI TEKNIK INFORMATIKA

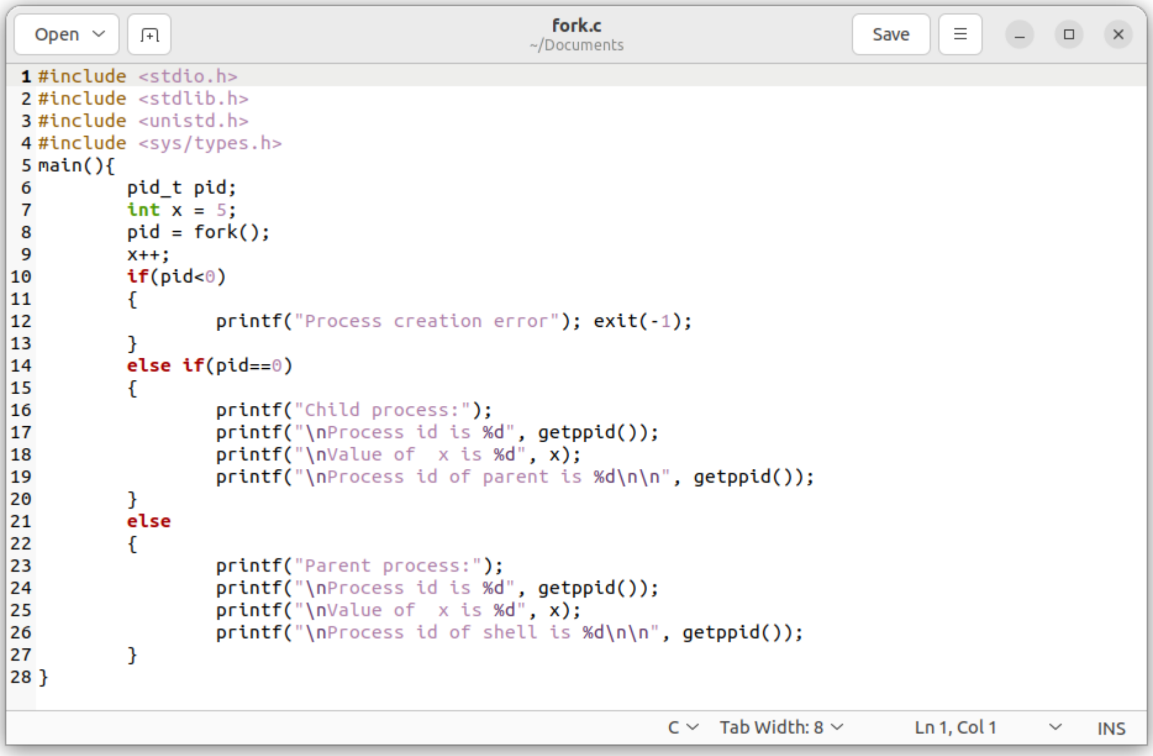
FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

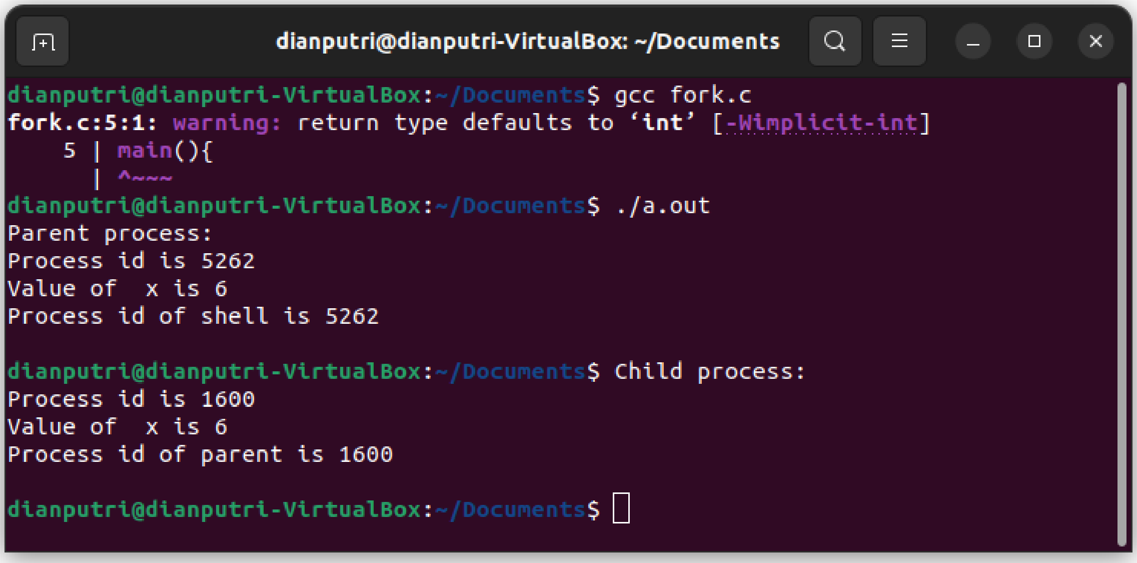
2021

Praktikum 1

Membuat sebuah ‘child process’ (proses baru) dengan menggunakan system call fork.



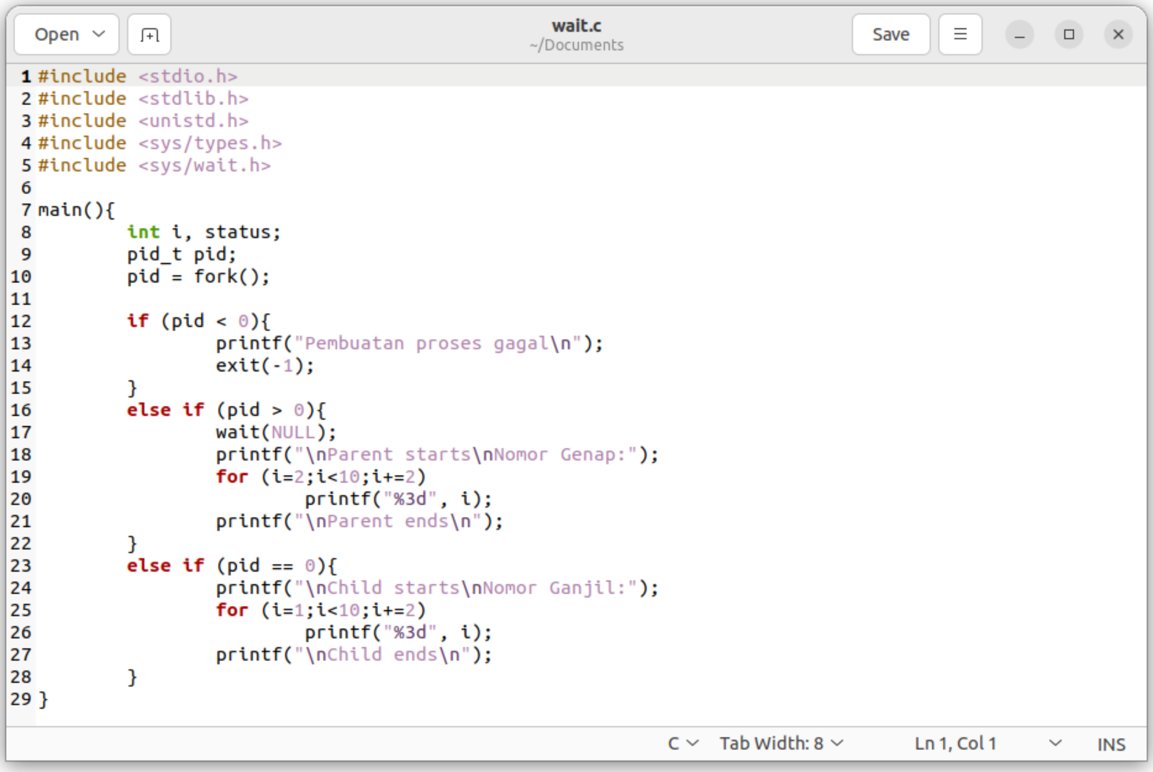
```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
main(){
	pid_t pid;
	int x = 5;
	pid = fork();
	x++;
	if(pid<0)
	{
		printf("Process creation error"); exit(-1);
	}
	else if(pid==0)
	{
		printf("Child process:");
		printf("\nProcess id is %d", getppid());
		printf("\nValue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



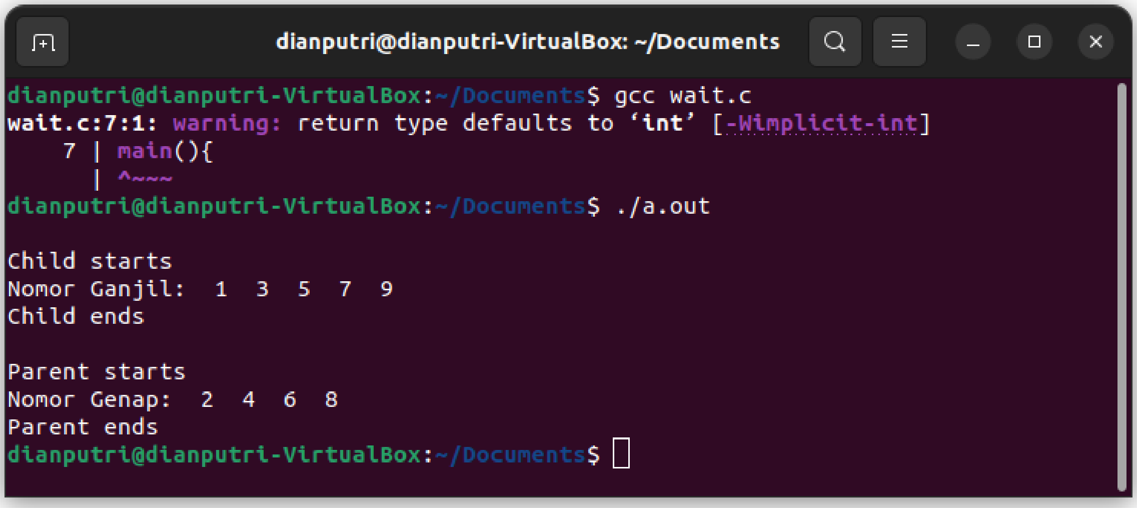
```
dianputri@dianputri-VirtualBox:~/Documents$ gcc fork.c
fork.c:5:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
    5 | main(){
      | ^~~~
dianputri@dianputri-VirtualBox:~/Documents$ ./a.out
Parent process:
Process id is 5262
Value of  x is 6
Process id of shell is 5262

dianputri@dianputri-VirtualBox:~/Documents$ Child process:
Process id is 1600
Value of  x is 6
Process id of parent is 1600

dianputri@dianputri-VirtualBox:~/Documents$ 
```

Praktikum 2

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



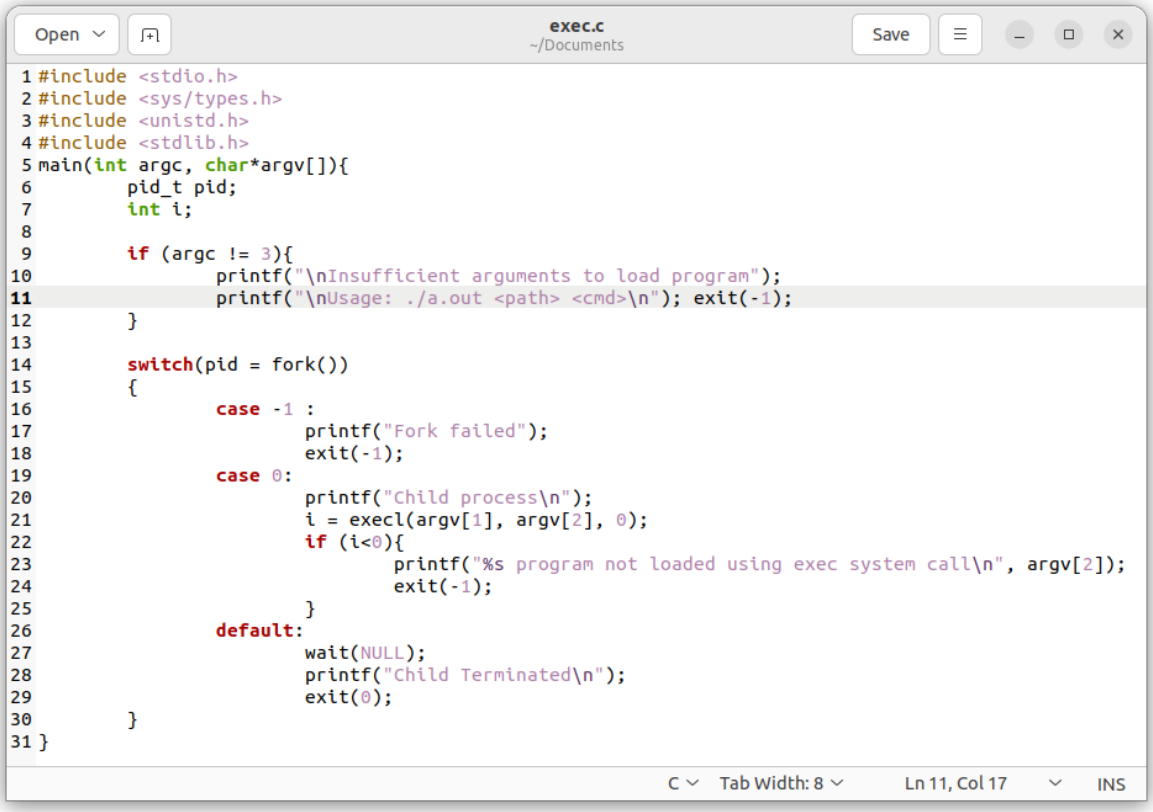
```c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

main(){
        int i, status;
        pid_t pid;
        pid = fork();

        if (pid < 0){
                printf("Pembuatan 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");
        }
}
```



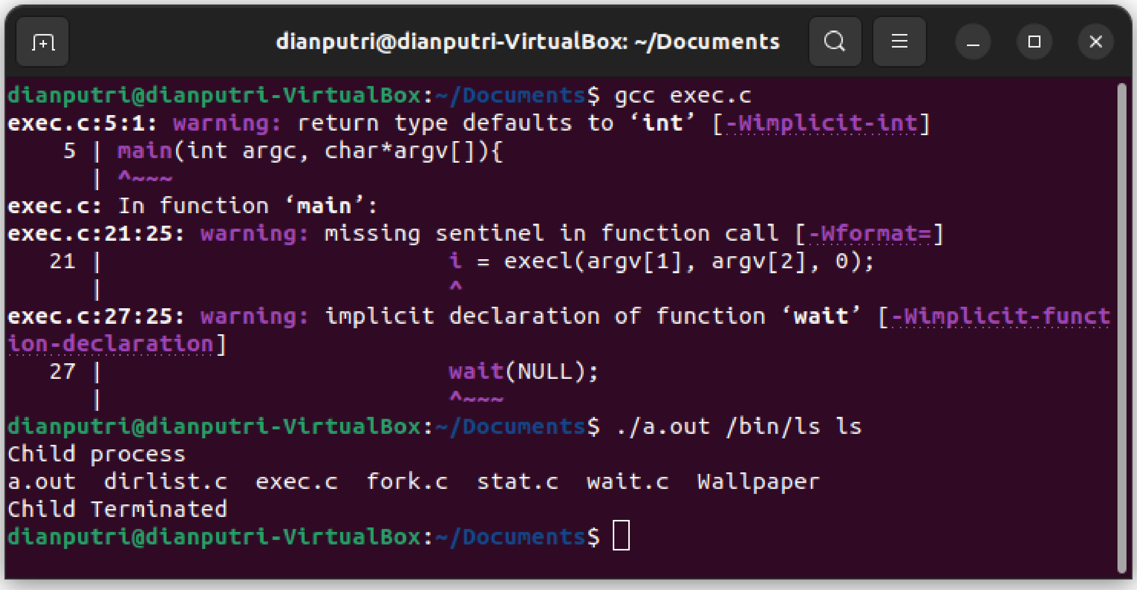
```
dianputri@dianputri-VirtualBox:~/Documents$ gcc wait.c
wait.c:7:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
    7 | main(){
      | ^~~~
dianputri@dianputri-VirtualBox:~/Documents$ ./a.out

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

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

Praktikum 3

Loading program yang dapat dieksekusi dalam sebuah ‘child’ proses menggunakan perintah system call ‘exec’.



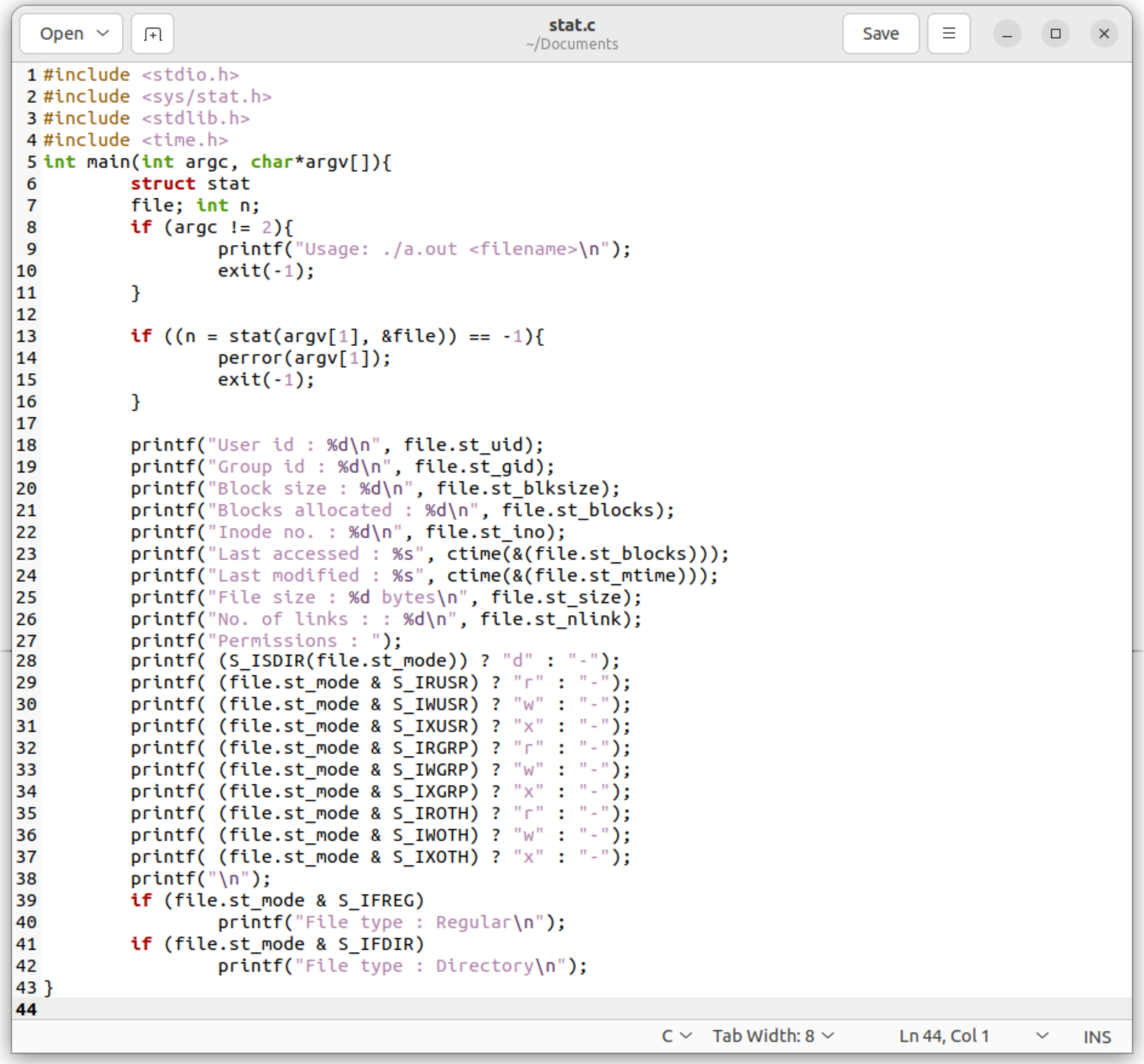
```c
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
main(int argc, char*argv[]){
        pid_t pid;
        int i;

        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]);
                                exit(-1);
                        }
                default:
                        wait(NULL);
                        printf("Child Terminated\n");
                        exit(0);
        }
}
```



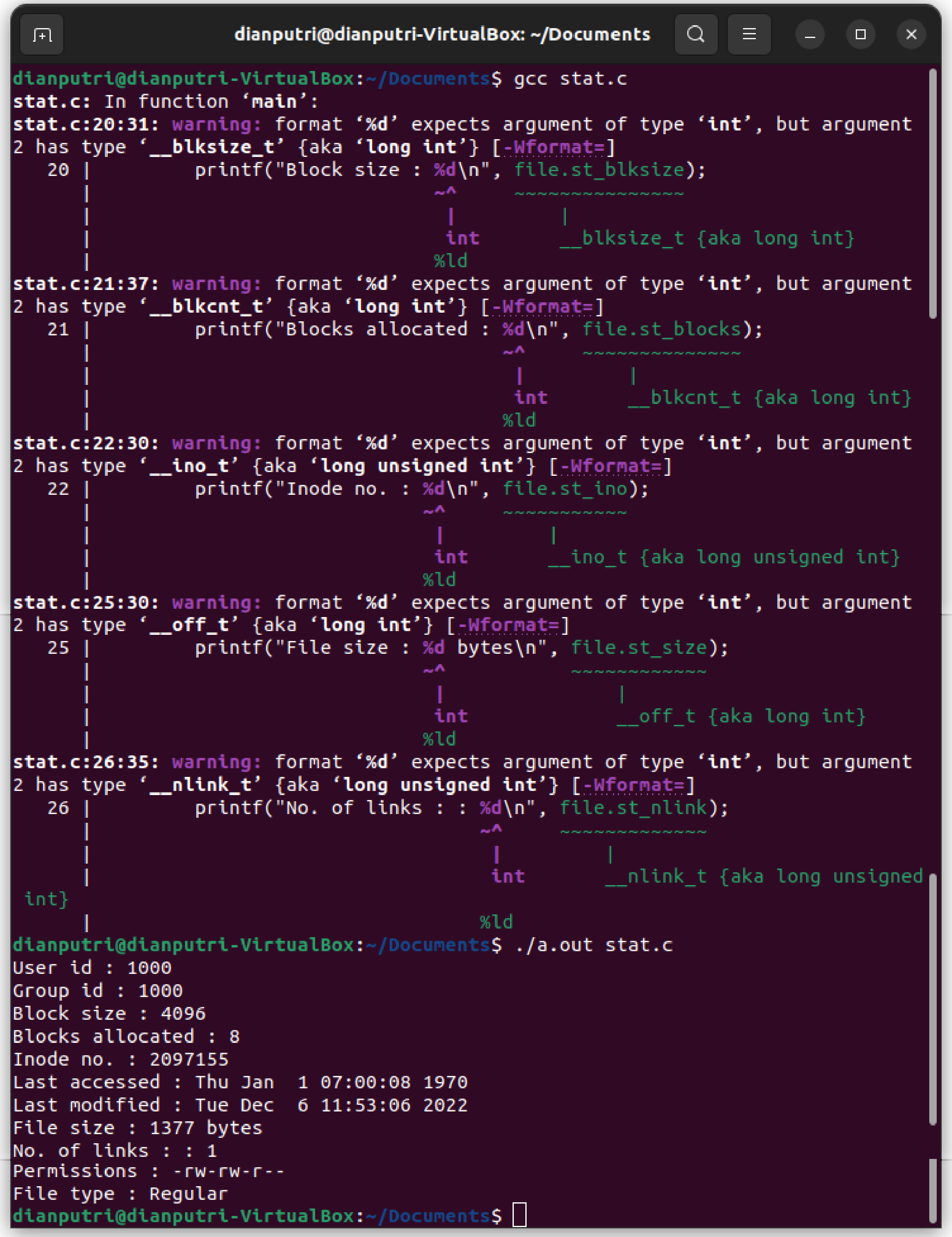
```
dianputri@dianputri-VirtualBox:~/Documents$ gcc exec.c
exec.c:5:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
    5 | main(int argc, char*argv[]){
      | ^~~~
exec.c: In function ‘main’:
exec.c:21:25: warning: missing sentinel in function call [-Wformat=]
   21 |                         i = execl(argv[1], argv[2], 0);
      |                         ^
exec.c:27:25: warning: implicit declaration of function ‘wait’ [-Wimplicit-funct
ion-declaration]
   27 |                         wait(NULL);
      |                         ^~~~
dianputri@dianputri-VirtualBox:~/Documents$ ./a.out /bin/ls ls
Child process
a.out  dirlist.c  exec.c  fork.c  stat.c  wait.c  Wallpaper
Child Terminated
dianputri@dianputri-VirtualBox:~/Documents$ 
```

Praktikum 4

Menampilkan status file menggunakan perintah system call ‘stat’.



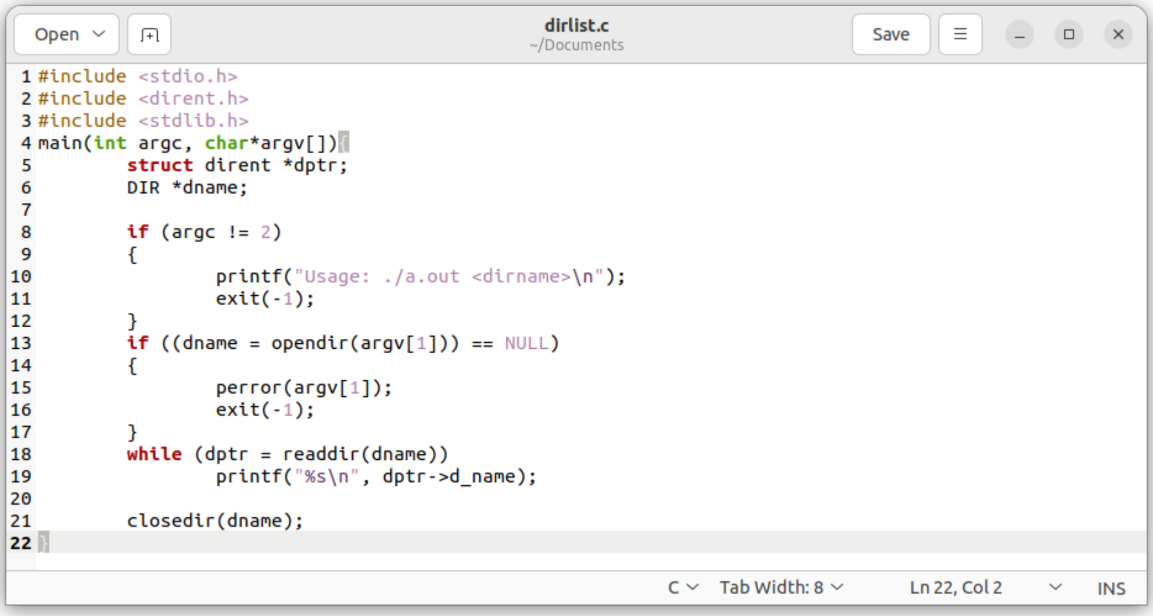
```c
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <time.h>
int main(int argc, char*argv[]){
        struct stat
        file; int n;
        if (argc != 2){
                printf("Usage: ./a.out <filename>\n");
                exit(-1);
        }

        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_blocks)));
        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");
}

```



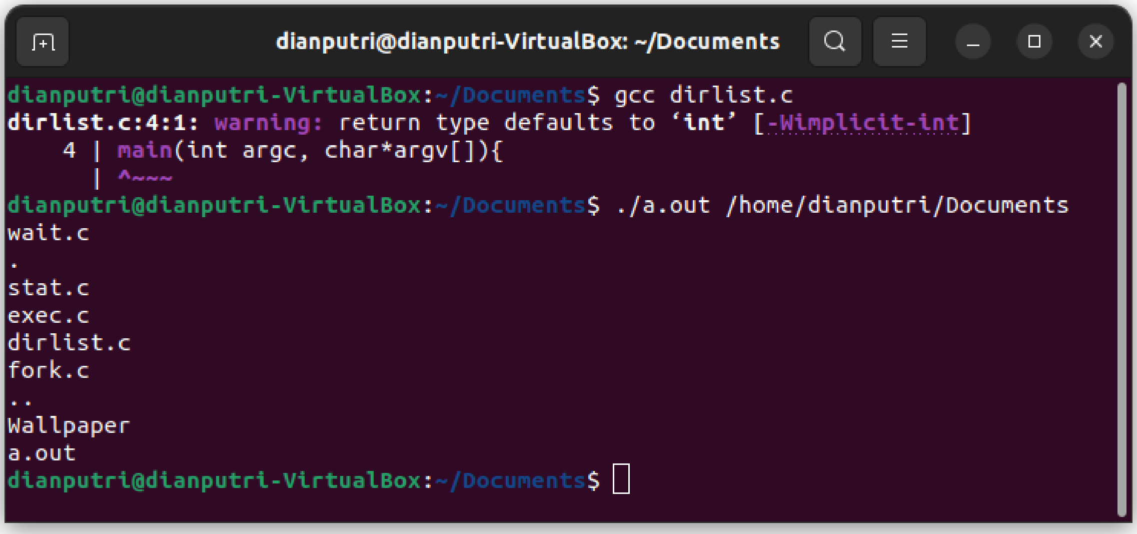
```
dianputri@dianputri-VirtualBox:~/Documents$ gcc stat.c
stat.c: In function ‘main’:
stat.c:20:31: warning: format ‘%d’ expects argument of type ‘int’, but argument
2 has type ‘__blksize_t’ {aka ‘long int’} [-Wformat=]
   20 |         printf("Block size : %d\n", file.st_blksize);
      |                              ~^     ~~~~~~~~~~~~~~~
      |                               |         |
      |                               int       __blksize_t {aka long int}
      |                              %ld
stat.c:21:37: warning: format ‘%d’ expects argument of type ‘int’, but argument
2 has type ‘__blkcnt_t’ {aka ‘long int’} [-Wformat=]
   21 |         printf("Blocks allocated : %d\n", file.st_blocks);
      |                                    ~^     ~~~~~~~~~~~~~~
      |                                     |         |
      |                                     int       __blkcnt_t {aka long int}
      |                                    %ld
stat.c:22:30: warning: format ‘%d’ expects argument of type ‘int’, but argument
2 has type ‘__ino_t’ {aka ‘long unsigned int’} [-Wformat=]
   22 |         printf("Inode no. : %d\n", file.st_ino);
      |                             ~^     ~~~~~~~~~~~
      |                              |         |
      |                              int       __ino_t {aka long unsigned int}
      |                             %ld
stat.c:25:30: warning: format ‘%d’ expects argument of type ‘int’, but argument
2 has type ‘__off_t’ {aka ‘long int’} [-Wformat=]
   25 |         printf("File size : %d bytes\n", file.st_size);
      |                             ~^           ~~~~~~~~~~~~
      |                              |               |
      |                              int             __off_t {aka long int}
      |                             %ld
stat.c:26:35: warning: format ‘%d’ expects argument of type ‘int’, but argument
2 has type ‘__nlink_t’ {aka ‘long unsigned int’} [-Wformat=]
   26 |         printf("No. of links : : %d\n", file.st_nlink);
      |                                  ~^     ~~~~~~~~~~~~~
      |                                   |         |
      |                                   int       __nlink_t {aka long unsigned
 int}
      |                                  %ld
dianputri@dianputri-VirtualBox:~/Documents$ ./a.out stat.c
User id : 1000
Group id : 1000
Block size : 4096
Blocks allocated : 8
Inode no. : 2097155
Last accessed : Thu Jan  1 07:00:08 1970
Last modified : Tue Dec  6 11:53:06 2022
File size : 1377 bytes
No. of links : : 1
Permissions : -rw-rw-r--
File type : Regular
dianputri@dianputri-VirtualBox:~/Documents$ 
```

Praktikum 5

Menampilkan isi direktori menggunakan perintah system call ‘readdir’.



```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);
}
```



```
dianputri@dianputri-VirtualBox:~/Documents$ gcc dirlist.c
dirlist.c:4:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
    4 | main(int argc, char*argv[]){
      | ^~~~
dianputri@dianputri-VirtualBox:~/Documents$ ./a.out /home/dianputri/Documents
wait.c
.
stat.c
exec.c
dirlist.c
fork.c
..
Wallpaper
a.out
dianputri@dianputri-VirtualBox:~/Documents$ 
```