PRAKTIKUM SISTEM OPERASI

MODUL 8

Disusun Oleh:

MOHAMAD HAFIZ SAPUTRO

L200210224

E

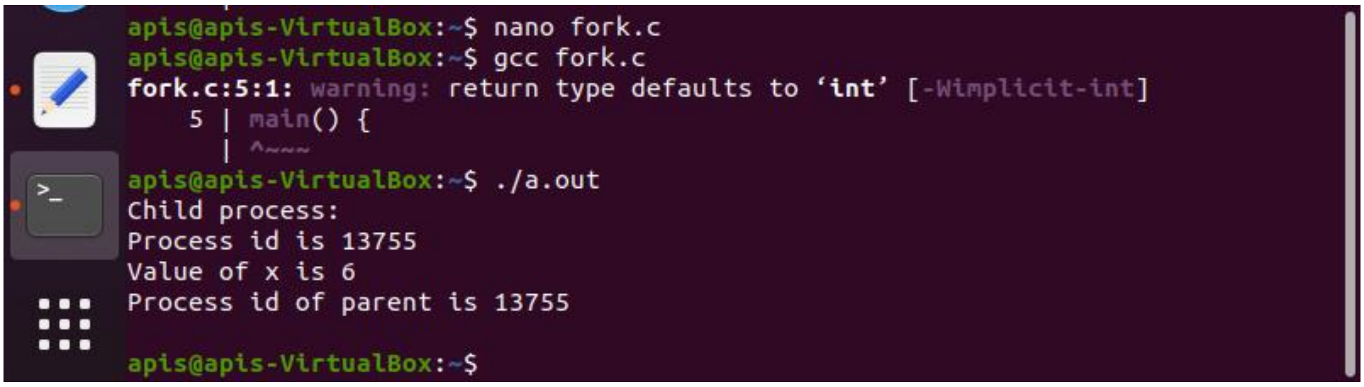
PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA

UNIVERSITAS MUHAMMADIYAH SURAKARTA

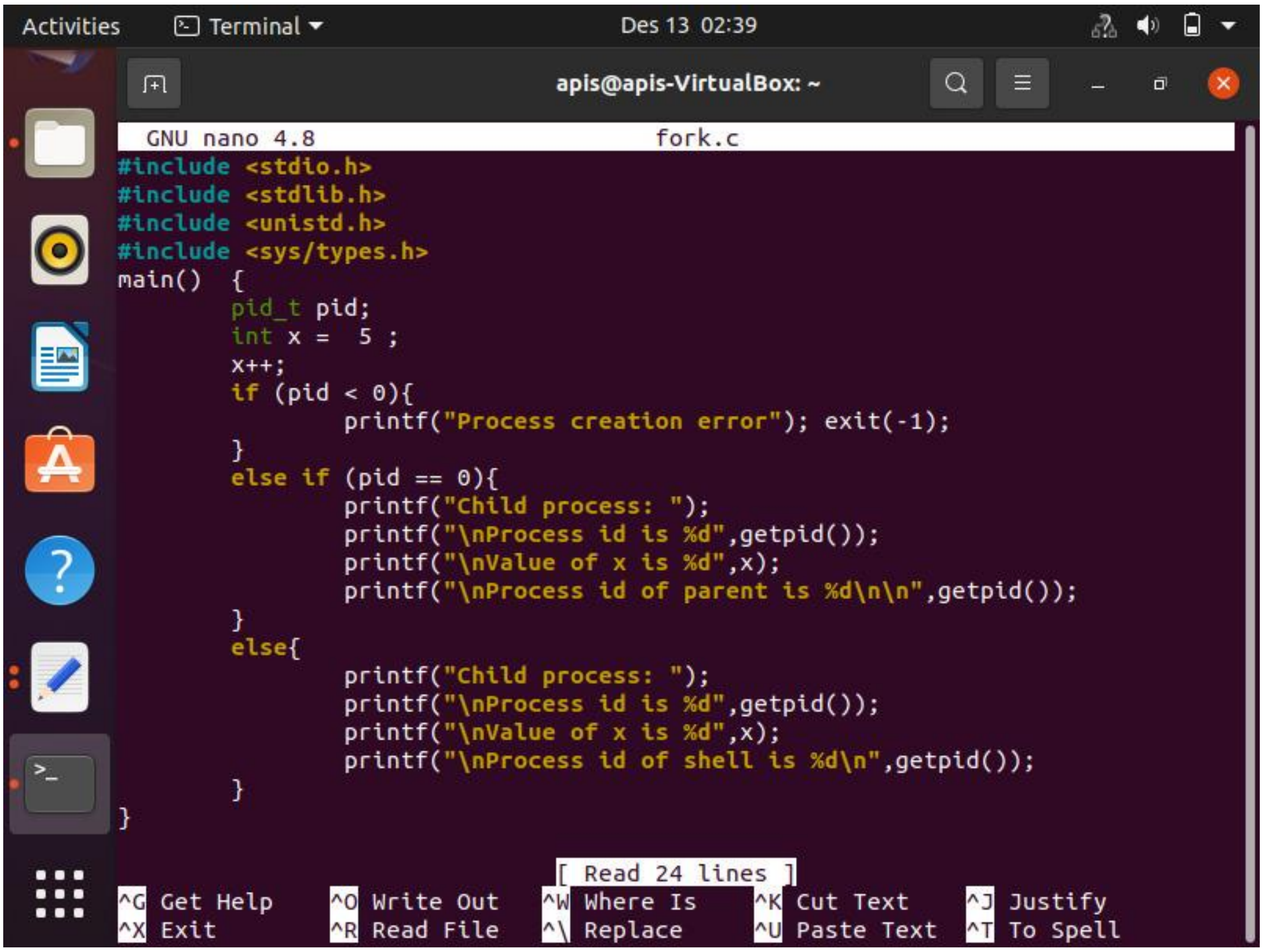
TAHUN 2022/2023

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



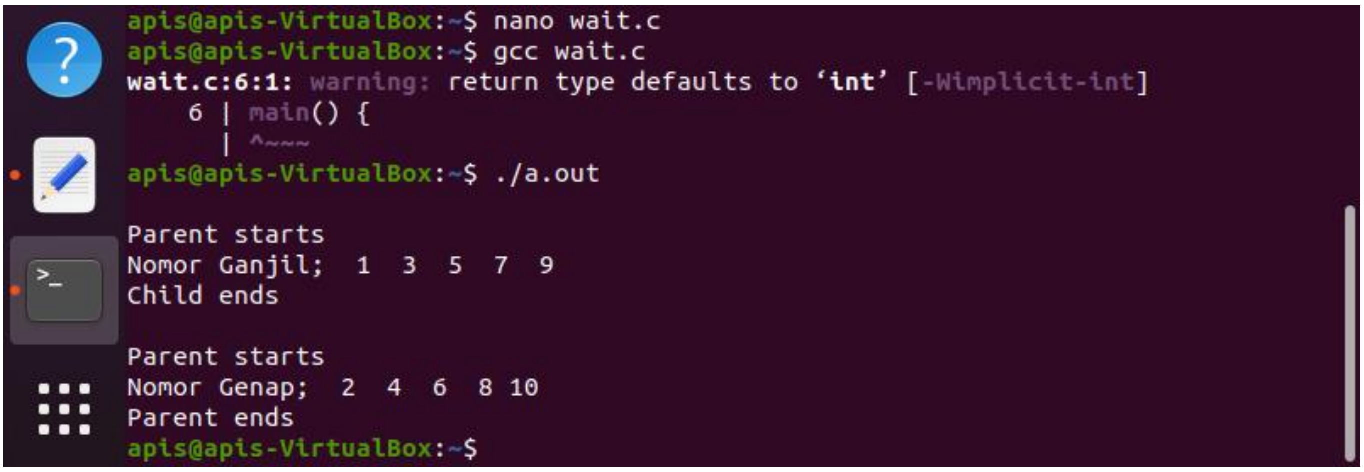
```
apis@apis-VirtualBox:~$ nano fork.c
apis@apis-VirtualBox:~$ gcc fork.c
fork.c:5:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
    5 | main() {
      | ^~~~
apis@apis-VirtualBox:~$ ./a.out
Child process:
Process id is 13755
Value of x is 6
Process id of parent is 13755

apis@apis-VirtualBox:~$
```

Fork.c code :



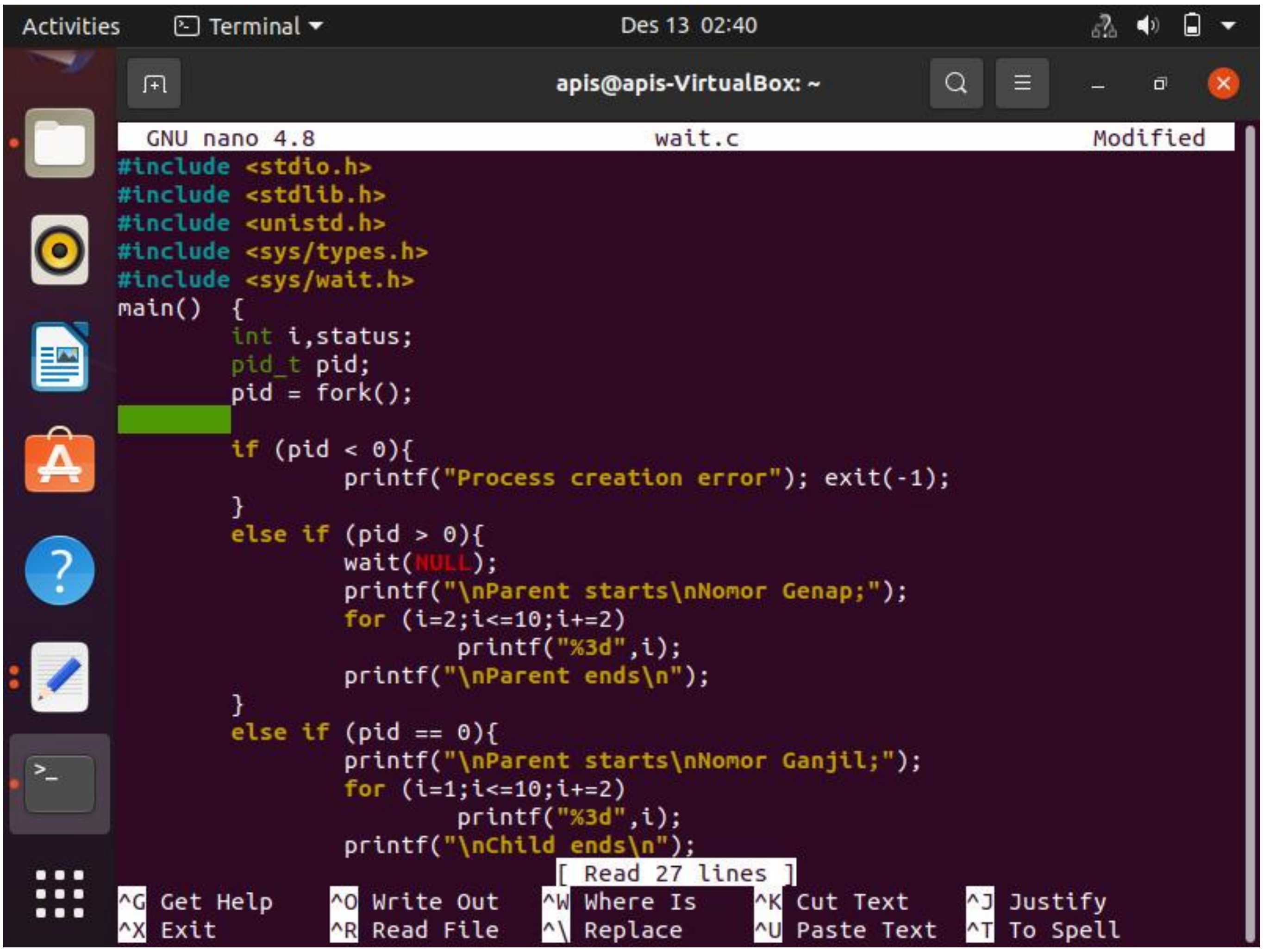
```
  GNU nano 4.8                       fork.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
main()  {
        pid_t pid;
        int x =  5 ;
        x++;
        if (pid < 0){
                printf("Process creation error"); exit(-1);
        }
        else if (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",getpid());
        }
        else{
                printf("Child process: ");
                printf("\nProcess id is %d",getpid());
                printf("\nValue of x is %d",x);
                printf("\nProcess id of shell is %d\n",getpid());
        }
}

                               [ Read 24 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell
```

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



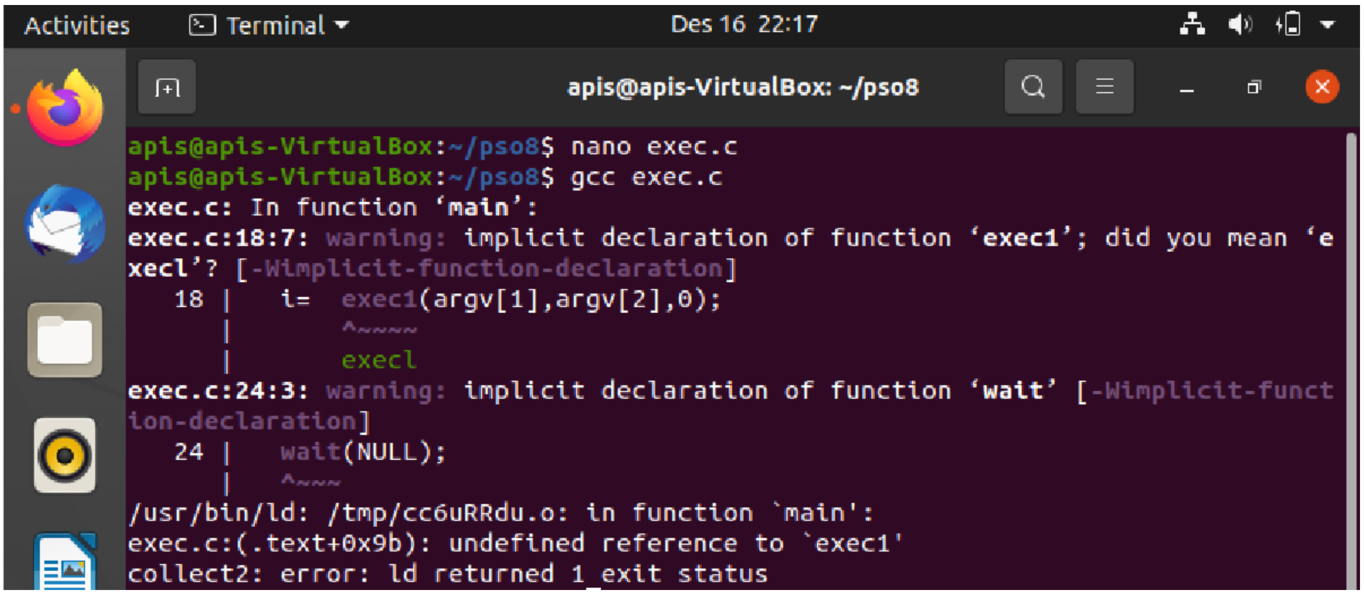
```
apis@apis-VirtualBox:~$ nano wait.c
apis@apis-VirtualBox:~$ gcc wait.c
wait.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
    6 | main() {
      | ^~~~
apis@apis-VirtualBox:~$ ./a.out

Parent starts
Nomor Ganjil;  1  3  5  7  9
Child ends

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

wait.c code



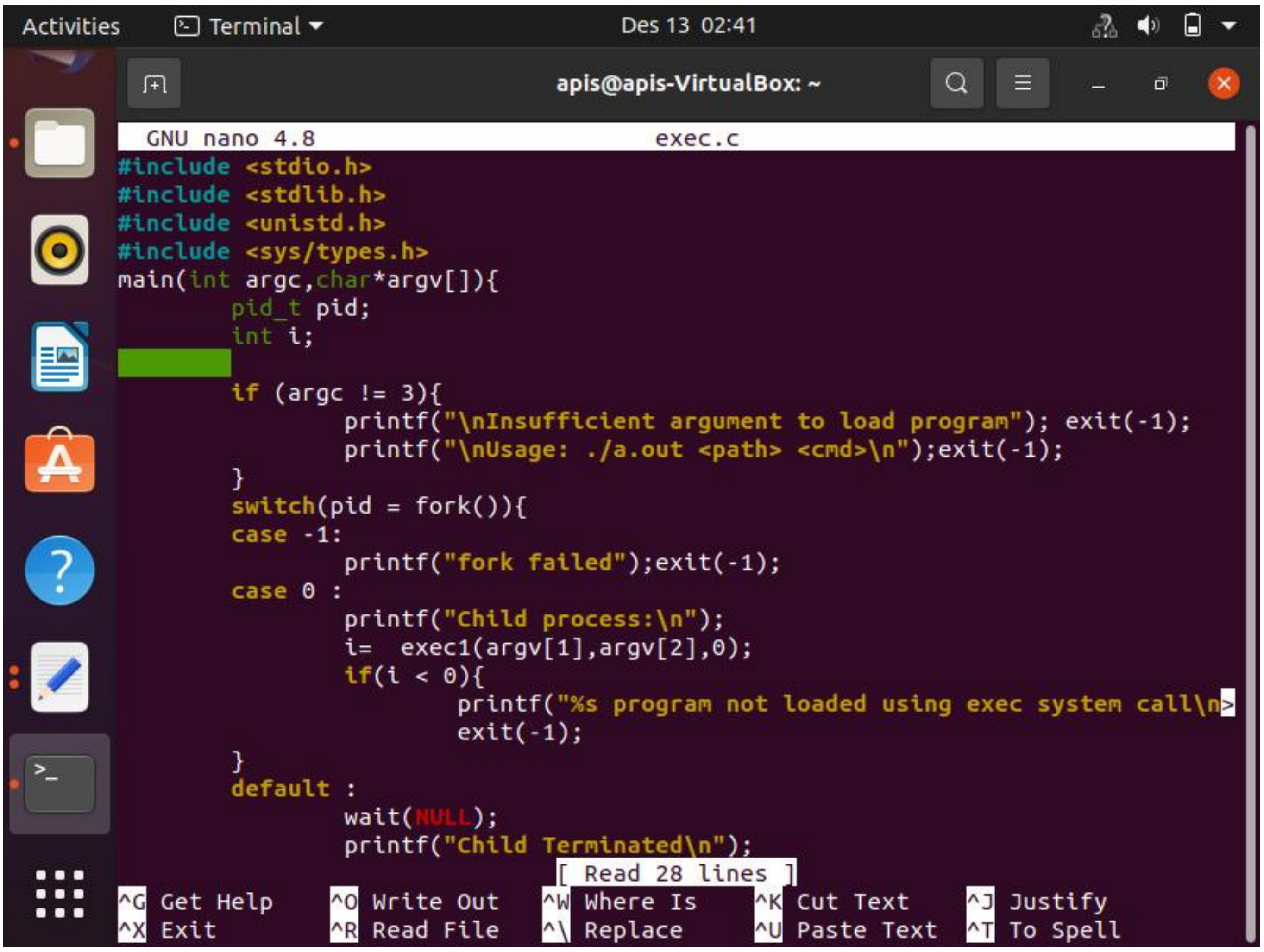
```
  GNU nano 4.8                    wait.c                          Modified
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
main()  {
        int i,status;
        pid_t pid;
        pid = fork();

        if (pid < 0){
                printf("Process creation error"); 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("\nParent starts\nNomor Ganjil;");
                for (i=1;i<=10;i+=2)
                        printf("%3d",i);
                printf("\nChild ends\n");
                              [ Read 27 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell
```

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



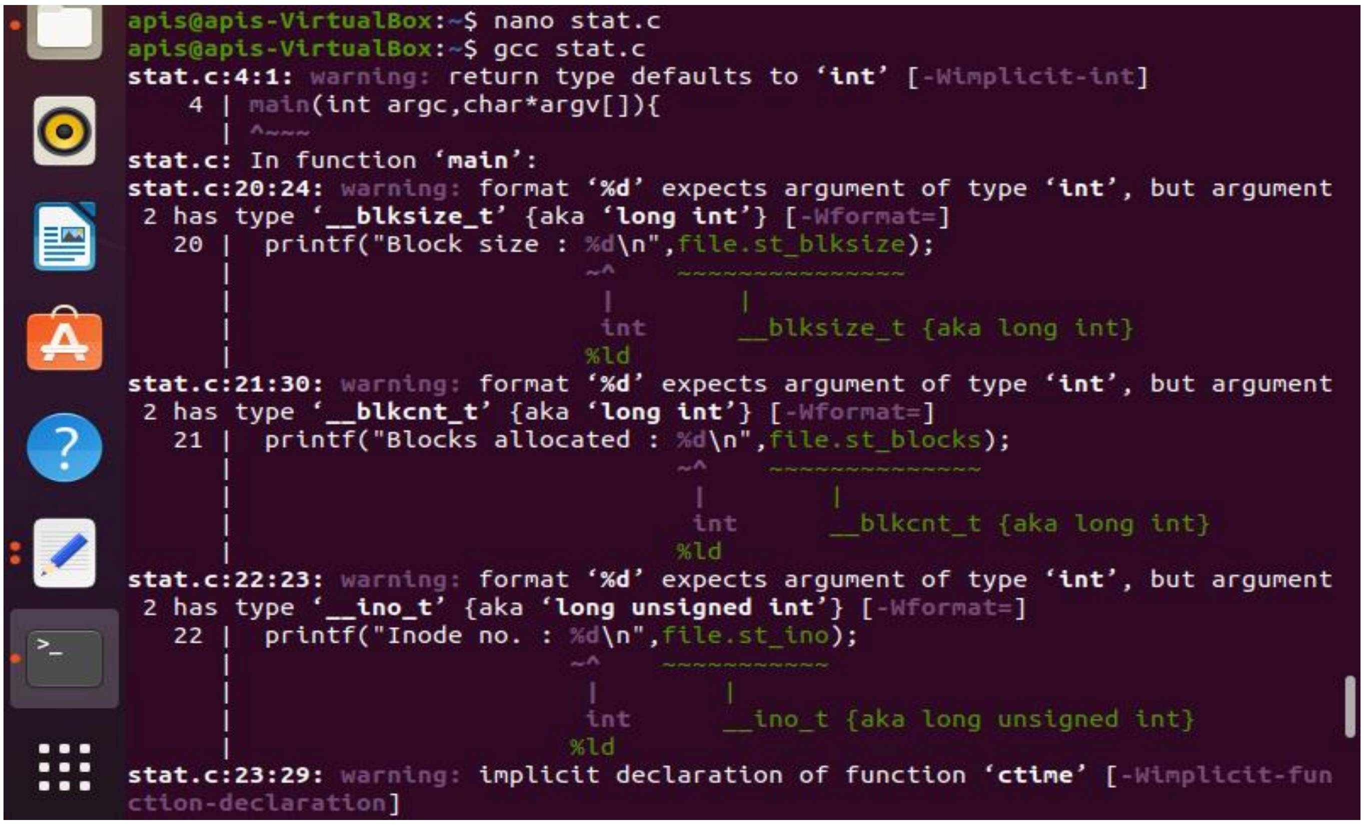
```
apis@apis-VirtualBox:~/pso8$ nano exec.c
apis@apis-VirtualBox:~/pso8$ gcc exec.c
exec.c: In function 'main':
exec.c:18:7: warning: implicit declaration of function 'execl'; did you mean 'e
xecl'? [-Wimplicit-function-declaration]
   18 |    i=  execl(argv[1],argv[2],0);
      |        ^~~~~
      |        execl
exec.c:24:3: warning: implicit declaration of function 'wait' [-Wimplicit-funct
ion-declaration]
   24 |    wait(NULL);
      |    ^~~~
/usr/bin/ld: /tmp/cc6uRRdu.o: in function `main':
exec.c:(.text+0x9b): undefined reference to `execl'
collect2: error: ld returned 1 exit status
```

exec.c code



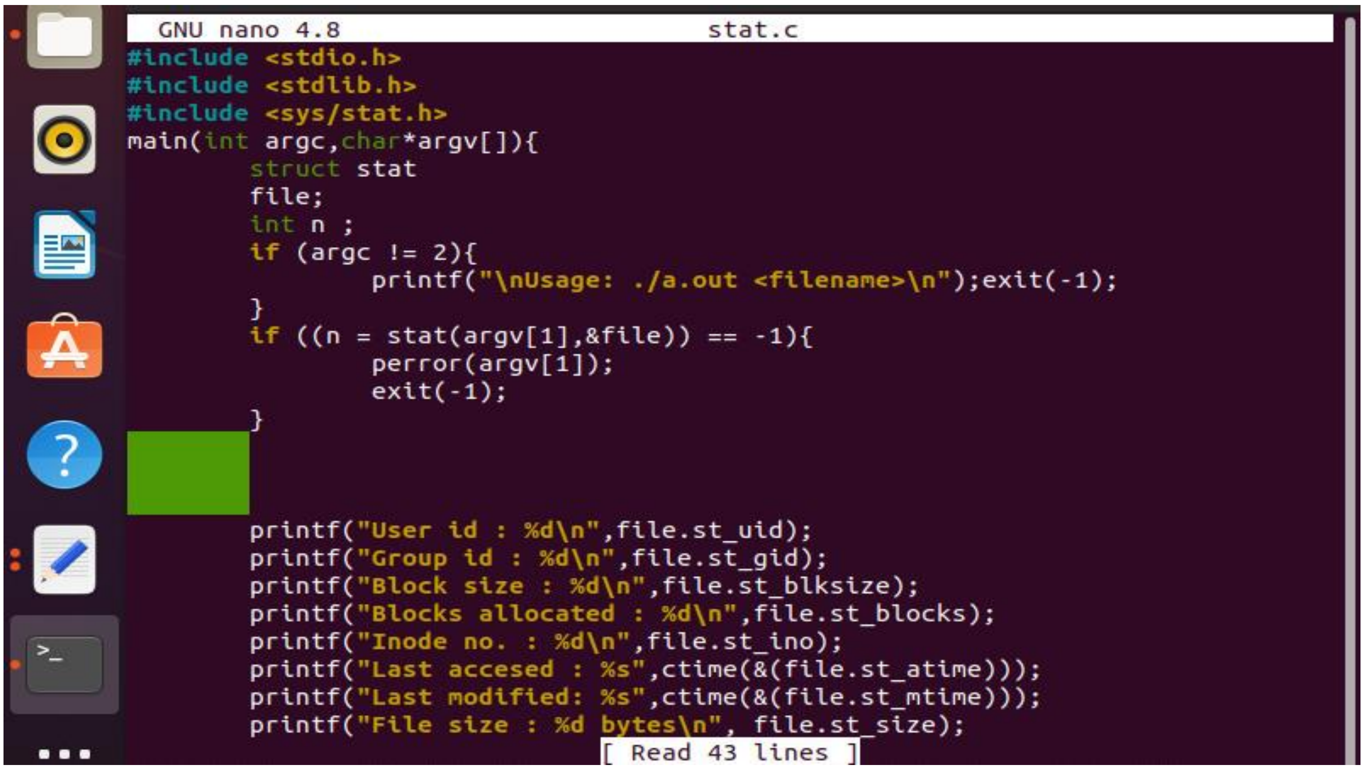
```
  GNU nano 4.8                        exec.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
main(int argc,char*argv[]){
        pid_t pid;
        int i;

        if (argc != 3){
                printf("\nInsufficient argument to load program"); exit(-1);
                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>
                        exit(-1);
        }
        default :
                wait(NULL);
                printf("Child Terminated\n");
                              [ Read 28 lines ]
^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify
^X Exit        ^R Read File   ^\ Replace     ^U Paste Text  ^T To Spell
```

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



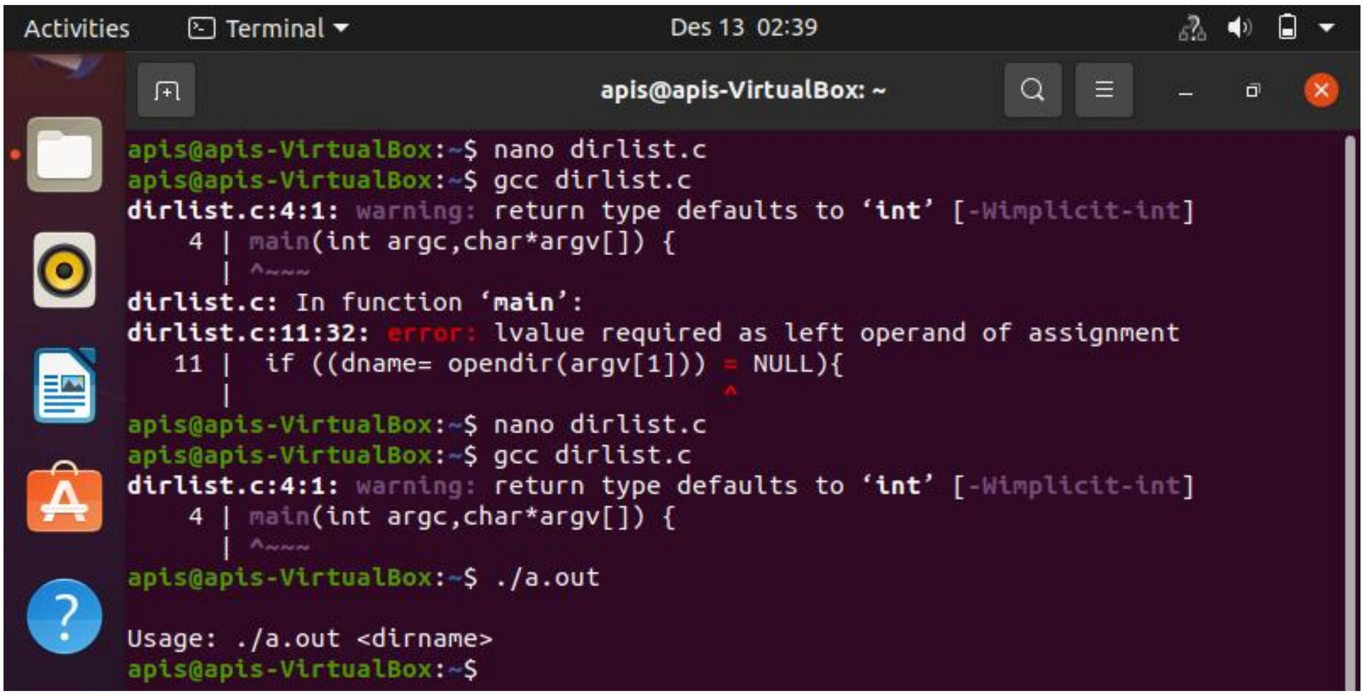
```
apis@apis-VirtualBox:~$ nano stat.c
apis@apis-VirtualBox:~$ gcc stat.c
stat.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
    4 | main(int argc,char*argv[]){
      | ^~~~
stat.c: In function 'main':
stat.c:20:24: warning: format '%d' expects argument of type 'int', but argument
 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:21:30: 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:22:23: 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:23:29: warning: implicit declaration of function 'ctime' [-Wimplicit-fun
ction-declaration]
```

Stat.c code



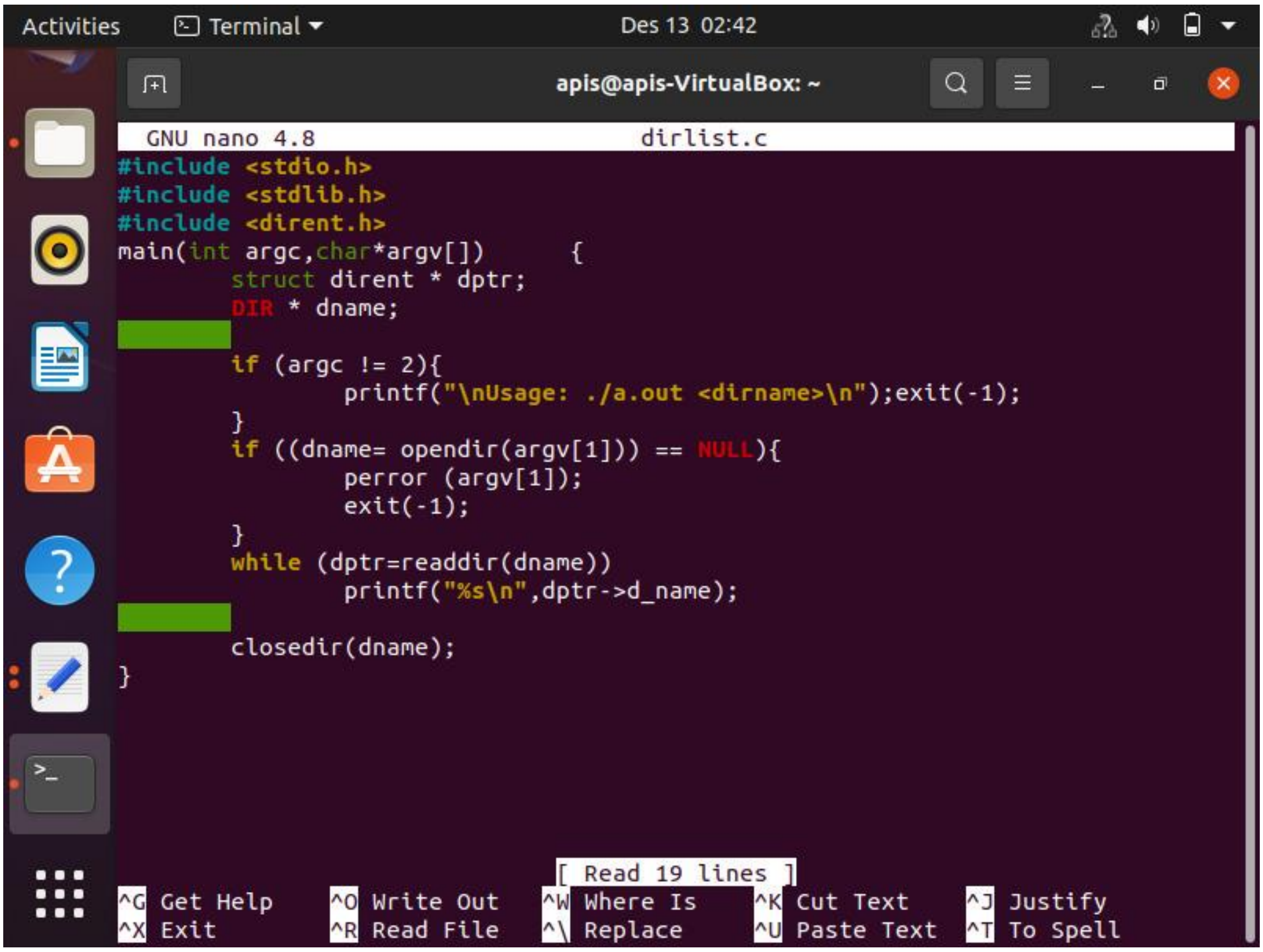
```
  GNU nano 4.8                          stat.c
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
main(int argc,char*argv[]){
        struct stat
        file;
        int n ;
        if (argc != 2){
                printf("\nUsage: ./a.out <filename>\n");exit(-1);
        }
        if ((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 accesed : %s",ctime(&(file.st_atime)));
        printf("Last modified: %s",ctime(&(file.st_mtime)));
        printf("File size : %d bytes\n", file.st_size);
                               [ Read 43 lines ]
```

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



```
apis@apis-VirtualBox:~$ nano dirlist.c
apis@apis-VirtualBox:~$ gcc dirlist.c
dirlist.c:4:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
    4 | main(int argc,char*argv[]) {
      | ^~~~
dirlist.c: In function ‘main’:
dirlist.c:11:32: error: lvalue required as left operand of assignment
   11 |  if ((dname= opendir(argv[1])) = NULL){
      |                                ^
apis@apis-VirtualBox:~$ nano dirlist.c
apis@apis-VirtualBox:~$ gcc dirlist.c
dirlist.c:4:1: warning: return type defaults to ‘int’ [-Wimplicit-int]
    4 | main(int argc,char*argv[]) {
      | ^~~~
apis@apis-VirtualBox:~$ ./a.out

Usage: ./a.out <dirname>
apis@apis-VirtualBox:~$
```

dirlist.c code



```
  GNU nano 4.8                        dirlist.c
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
main(int argc,char*argv[])      {
        struct dirent * dptr;
        DIR * dname;

        if (argc != 2){
                printf("\nUsage: ./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 19 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify
^X Exit         ^R Read File    ^\ Replace      ^U Paste Text   ^T To Spell
```