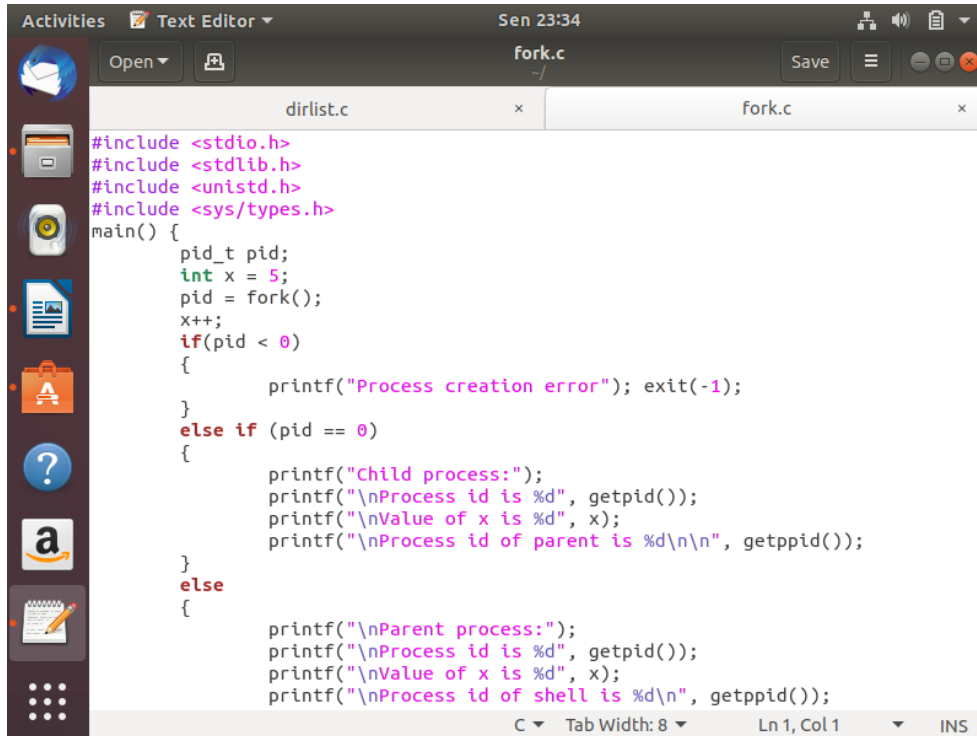


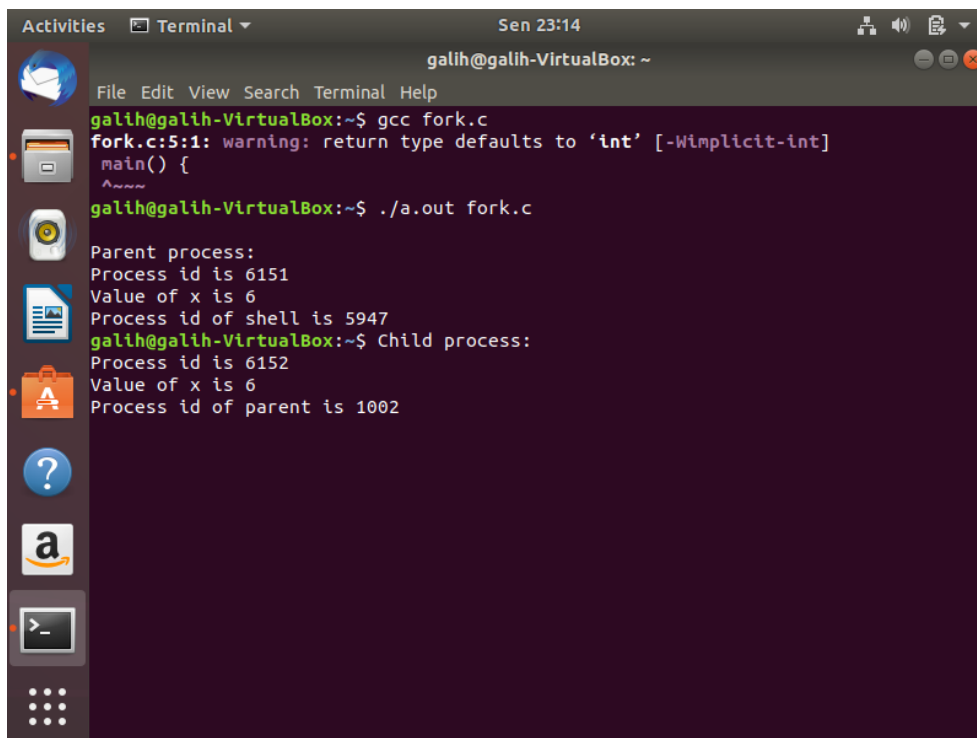
Nama : Galih Prayoga

NIM : L20018000

1. Fork.c



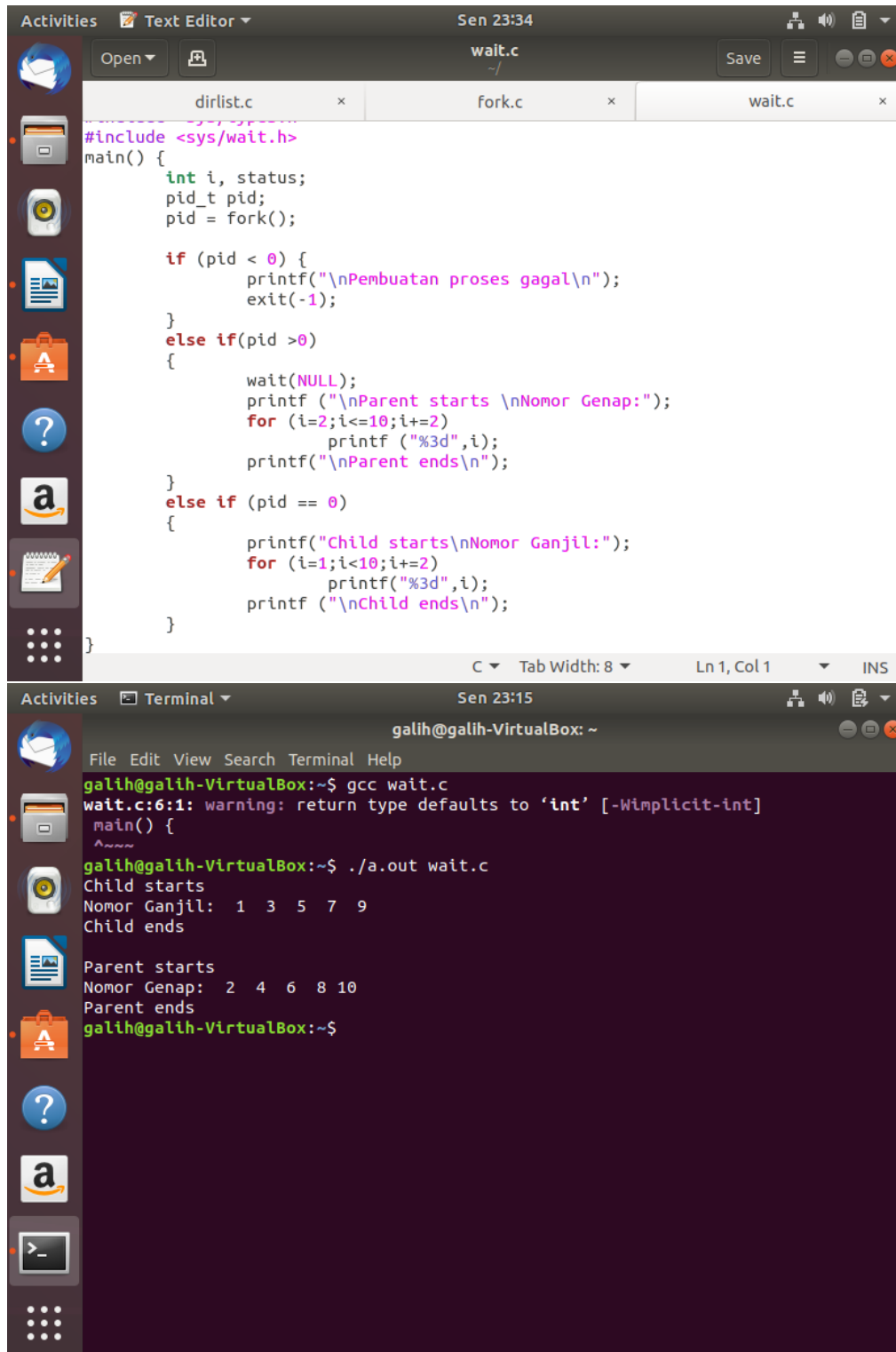
```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
main() {
    pid_t pid;
    int x = 5;
    pid = fork();
    x++;
    if(pid < 0)
    {
        printf("Process creation error"); exit(-1);
    }
    else if (pid == 0)
    {
        printf("Child process:");
     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



```
galih@galih-VirtualBox: ~
File Edit View Search Terminal Help
galih@galih-VirtualBox:~$ gcc fork.c
fork.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
main() {
^~~~~~
galih@galih-VirtualBox:~$ ./a.out fork.c

Parent process:
Process id is 6151
Value of x is 6
Process id of shell is 5947
galih@galih-VirtualBox:~$ Child process:
Process id is 6152
Value of x is 6
Process id of parent is 1002
```

2. Wait.c



The screenshot shows a Linux desktop environment with a sidebar on the left containing application icons. The main window is a text editor titled 'Text Editor' with a tab for 'wait.c'. The code in the editor is as follows:

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

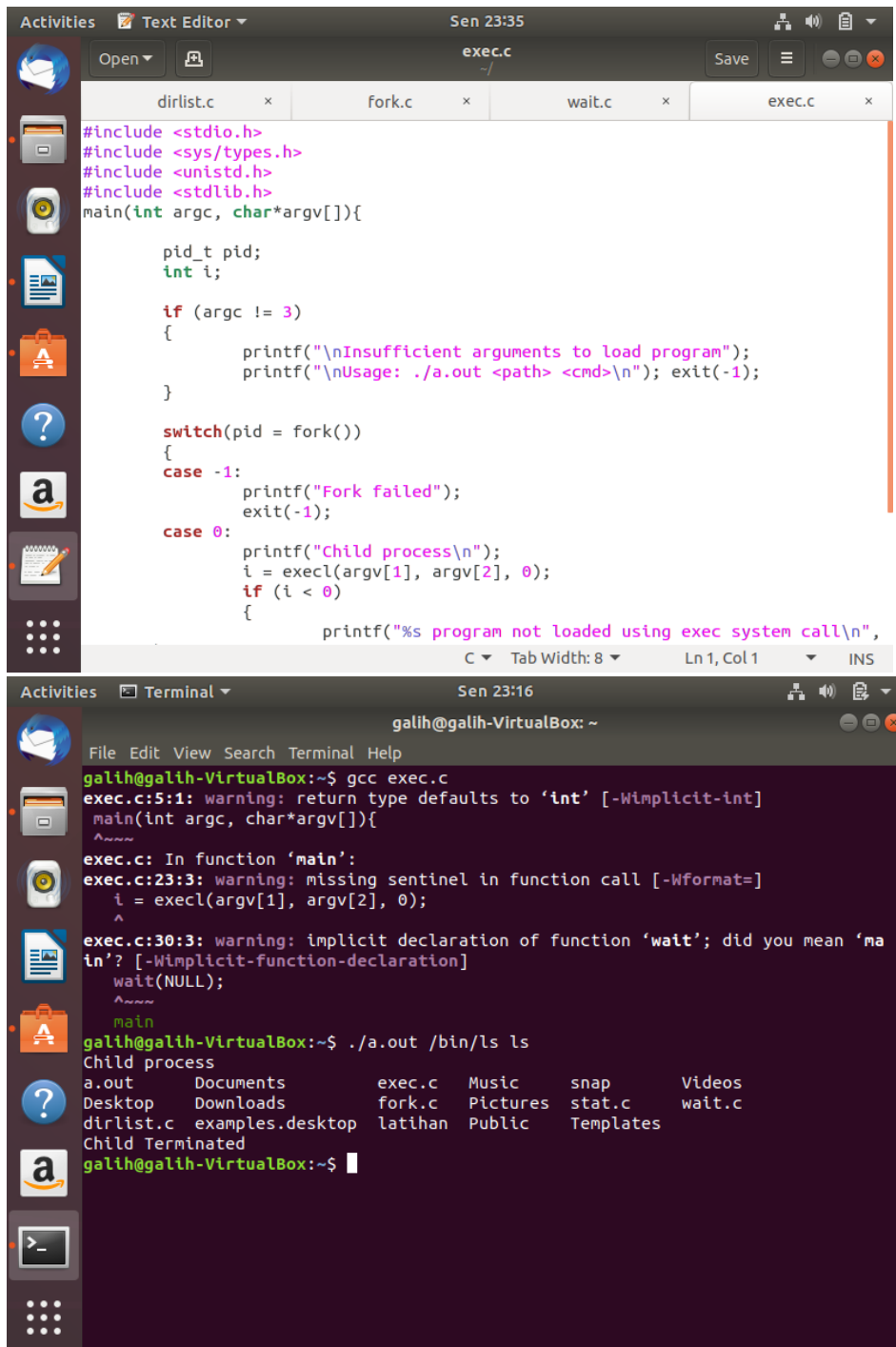
    if (pid < 0) {
        printf("\nPembuatan proses gagal\n");
        exit(-1);
    }
    else if (pid > 0)
    {
        wait(NULL);
        printf ("\nParent starts \nNomor Genap:");
        for (i=2;i<=10;i+=2)
            printf ("%3d",i);
        printf("\nParent ends\n");
    }
    else if (pid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

Below the text editor is a terminal window titled 'Terminal' with the prompt 'galih@galih-VirtualBox: ~'. The terminal shows the following commands and output:

```
galih@galih-VirtualBox:~$ gcc wait.c
wait.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
main() {
^~~~~
galih@galih-VirtualBox:~$ ./a.out wait.c
Child starts
Nomor Ganjil:  1  3  5  7  9
Child ends

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

3. Exec.c



The screenshot shows a Linux desktop environment with a text editor and a terminal window. The text editor, titled "Text Editor", displays the source code for a C program named "exec.c". The code includes headers for `<stdio.h>`, `<sys/types.h>`, `<unistd.h>`, and `<stdlib.h>`. The `main` function takes `argc` and `argv` as arguments. It checks if `argc` is not equal to 3, and if so, it prints an error message and exits. Otherwise, it forks a child process. In the child process, it prints "Child process", calls `execl` with `argv[1]`, `argv[2]`, and `0`, and then checks if the return value is less than 0. If so, it prints a message indicating that the program was not loaded using the `exec` system call. The terminal window, titled "Terminal", shows the compilation of `exec.c` using `gcc`. It displays several warnings: "warning: return type defaults to 'int' [-Wimplicit-int]", "warning: missing sentinel in function call [-Wformat=]", and "warning: implicit declaration of function 'wait'; did you mean 'main'? [-Wimplicit-function-declaration]". The terminal also shows the execution of the program using `./a.out /bin/ls ls`, which results in the output of `ls` being displayed. The output shows the contents of the current directory, including `a.out`, `Documents`, `exec.c`, `Music`, `snap`, `Videos`, `Desktop`, `Downloads`, `fork.c`, `Pictures`, `stat.c`, `wait.c`, `dirlist.c`, `examples.desktop`, `latihan`, `Public`, and `Templates`. The terminal also shows the message "Child Terminated" and the prompt `galih@galih-VirtualBox:~$`.

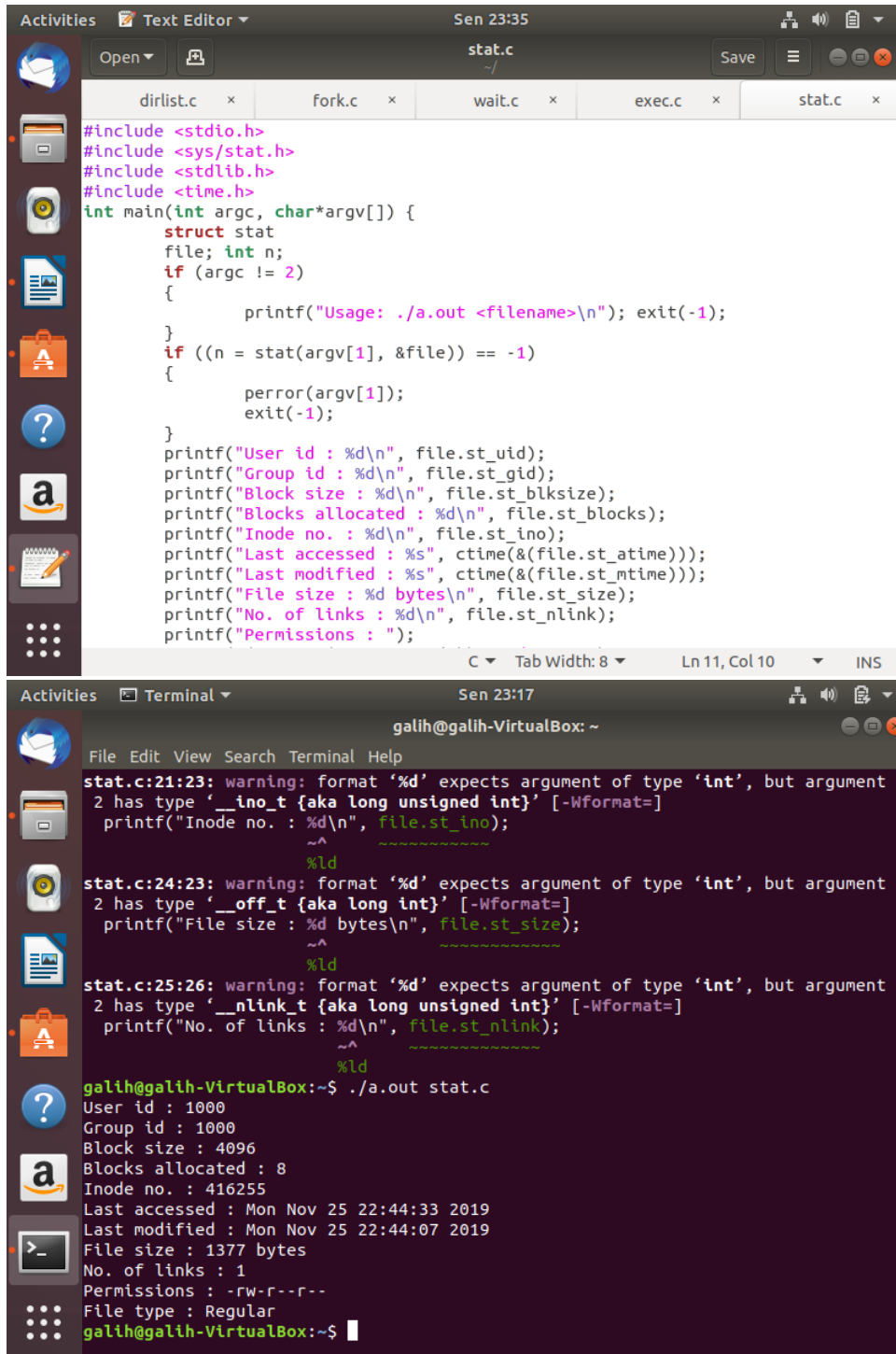
```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
main(int argc, char*argv[]){
    pid_t pid;
    int i;

    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",
                    C Tab Width: 8 Ln 1, Col 1 INS
                );
            }
    }
}
```

```
galih@galih-VirtualBox: ~
File Edit View Search Terminal Help
galih@galih-VirtualBox:~$ gcc exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char*argv[]){
^
exec.c: In function 'main':
exec.c:23:3: warning: missing sentinel in function call [-Wformat=]
    i = execl(argv[1], argv[2], 0);
    ^
exec.c:30:3: warning: implicit declaration of function 'wait'; did you mean 'ma
in'? [-Wimplicit-function-declaration]
    wait(NULL);
    ^
main
galih@galih-VirtualBox:~$ ./a.out /bin/ls ls
Child process
a.out      Documents      exec.c      Music       snap        Videos
Desktop    Downloads      fork.c      Pictures    stat.c      wait.c
dirlist.c  examples.desktop  latihan     Public      Templates
Child Terminated
galih@galih-VirtualBox:~$
```

4. Stat.c



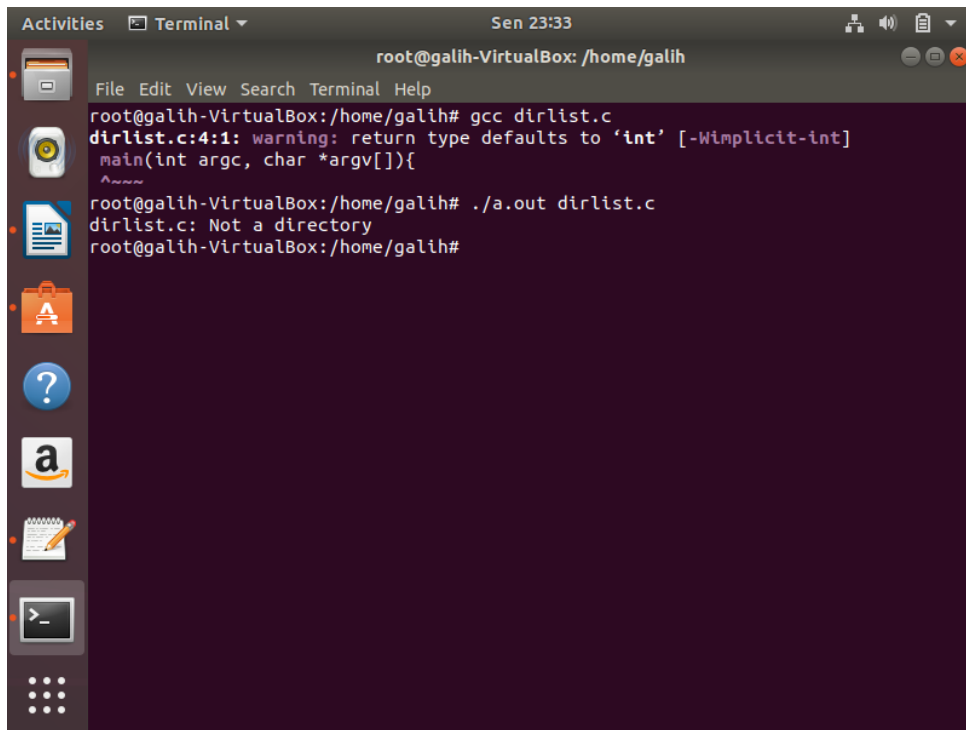
The image shows a Linux desktop environment with a text editor and a terminal window. The text editor, titled 'stat.c', contains the following C code:

```
#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 : ");
```

The terminal window shows the compilation and execution of the program. It displays several warnings about format specifiers and the final output of the program when run with the argument 'stat.c':

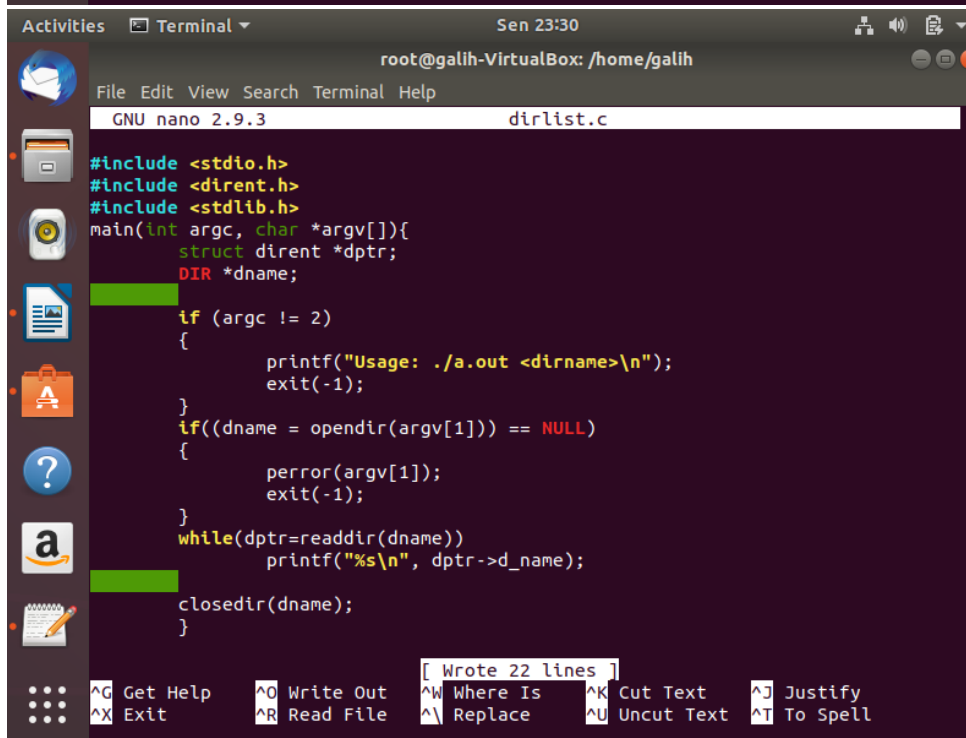
```
stat.c:21:23: warning: format '%d' expects argument of type 'int', but argument
2 has type '__ino_t {aka long unsigned int}' [-Wformat=]
    printf("Inode no. : %d\n", file.st_ino);
                           ^~
                           %ld
stat.c:24:23: warning: format '%d' expects argument of 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:25:26: 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
galih@galih-VirtualBox:~$ ./a.out stat.c
User id : 1000
Group id : 1000
Block size : 4096
Blocks allocated : 8
Inode no. : 416255
Last accessed : Mon Nov 25 22:44:33 2019
Last modified : Mon Nov 25 22:44:07 2019
File size : 1377 bytes
No. of links : 1
Permissions : -rw-r--r--
File type : Regular
galih@galih-VirtualBox:~$
```

5. Dirlist.c



A terminal window titled "root@galih-VirtualBox: /home/galih" showing the compilation and execution of a C program named dirlist.c. The user runs 'gcc dirlist.c', which produces a warning about the return type of 'main' defaulting to 'int'. Then, the user runs './a.out dirlist.c', which results in an error: 'dirlist.c: Not a directory'.

```
root@galih-VirtualBox: /home/galih
File Edit View Search Terminal Help
root@galih-VirtualBox:/home/galih# gcc dirlist.c
dirlist.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char *argv[]){
^
root@galih-VirtualBox:/home/galih# ./a.out dirlist.c
dirlist.c: Not a directory
root@galih-VirtualBox:/home/galih#
```



A terminal window titled "root@galih-VirtualBox: /home/galih" showing the source code of dirlist.c in the nano editor. The code includes headers for stdio, dirent, and stdlib, and implements a main function that checks the number of arguments, prints usage, and lists the contents of a directory using opendir and readdir.

```
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);
}
```

[Wrote 22 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify
^X Exit	^R Read File	^_ Replace	^U Uncut Text	^T To Spell