

PRAKTIKUM SISTEM OPERASI
MODUL 8



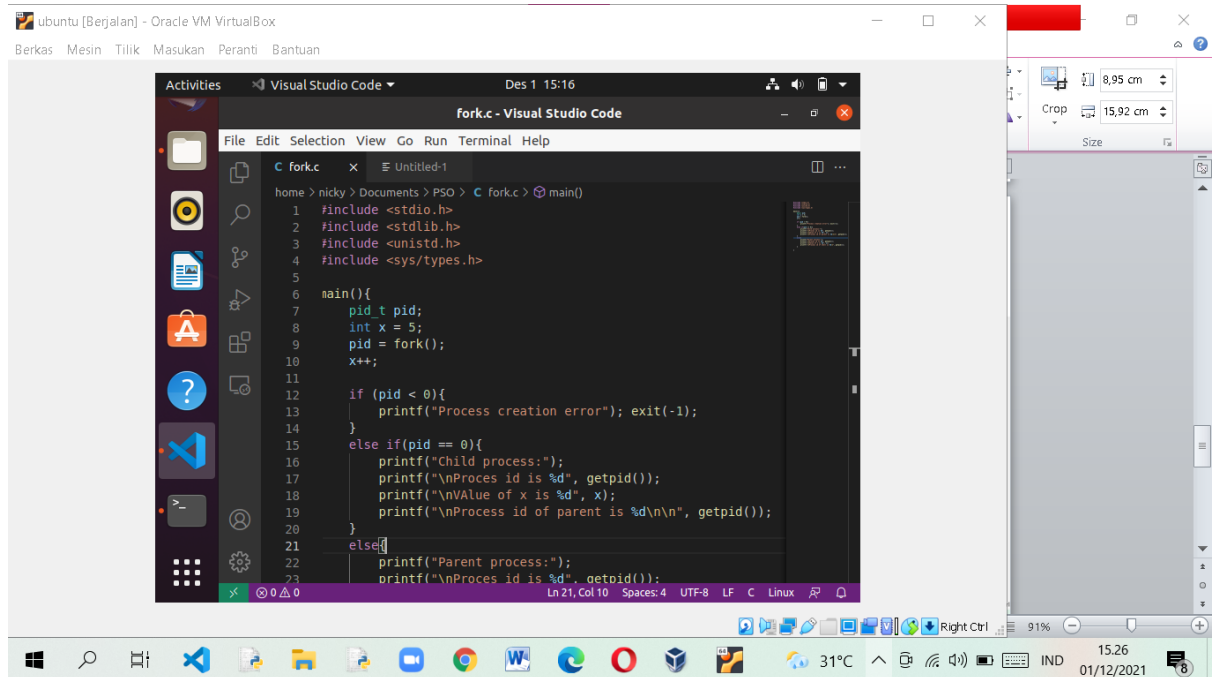
Nama : NICKY JULYATRIKA SARI

NIM : L200200101

PROGRAM STUDI
INFORMATIKA

FAKULTAS KOMUNIKASI DAN INFORMATIKA
UNIVERSITAS MUHAMMADIYAH SURAKARTA
TAHUN 2021/2022

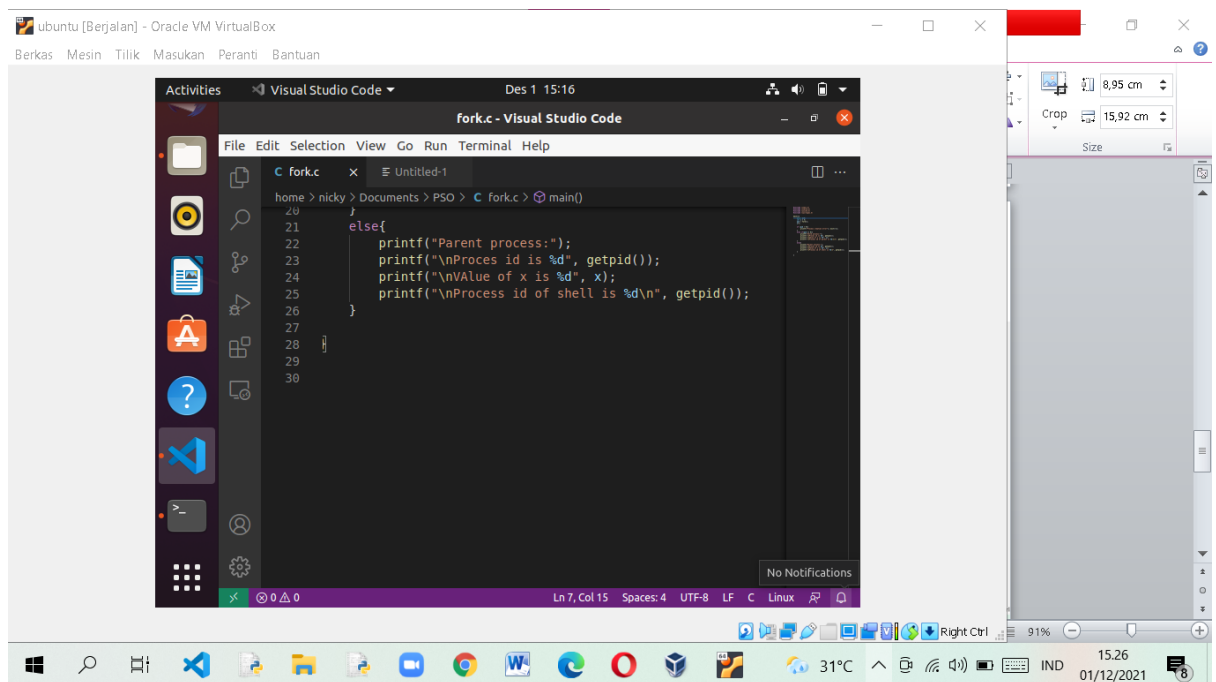
1. Membuat sebuah 'child process' (proses baru) dengan menggunakan system call 'fork'



The screenshot shows the Visual Studio Code editor with a C file named `fork.c`. The code is as follows:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5
6 main(){
7     pid_t pid;
8     int x = 5;
9     pid = fork();
10    x++;
11
12    if (pid < 0){
13        printf("Process creation error"); exit(-1);
14    }
15    else if(pid == 0){
16        printf("Child process:");
17        printf("\nProces id is %d", getpid());
18        printf("\nValue of x is %d", x);
19        printf("\nProcess id of parent is %d\n", getpid());
20    }
21    else{
22        printf("Parent process:");
23        printf("\nProces id is %d", getpid());
```

The status bar at the bottom indicates the cursor is at line 21, column 10.



The screenshot shows the Visual Studio Code editor with the same `fork.c` file, but with the code completed. The code is as follows:

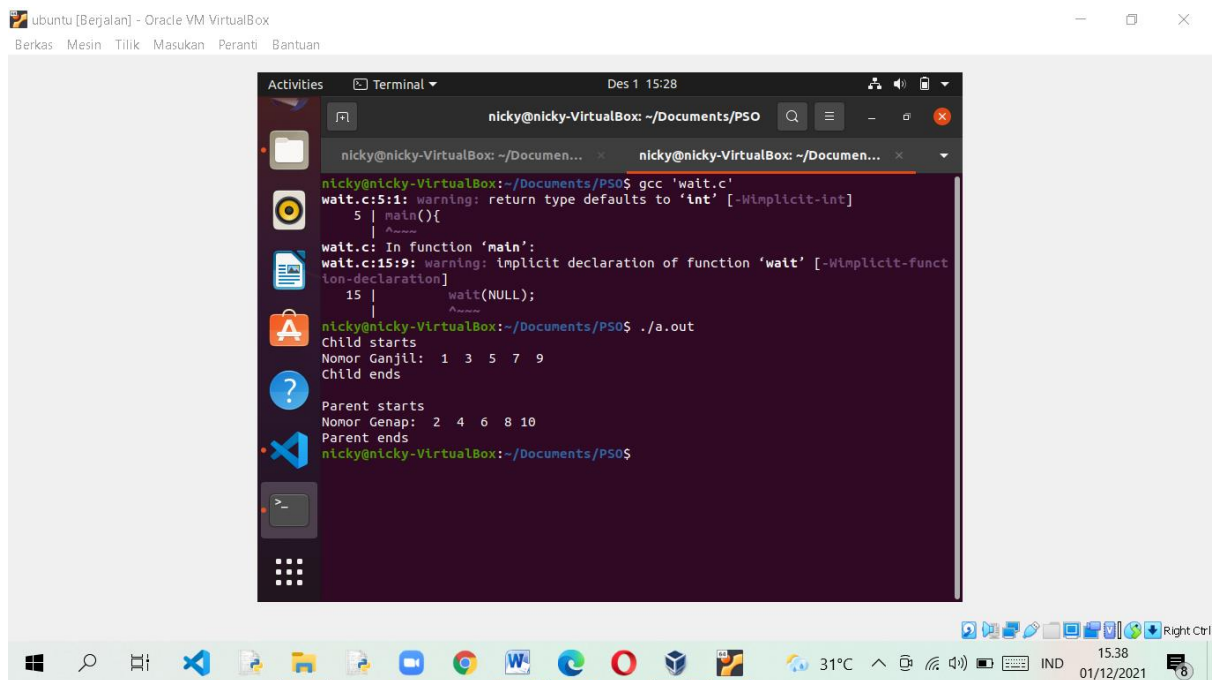
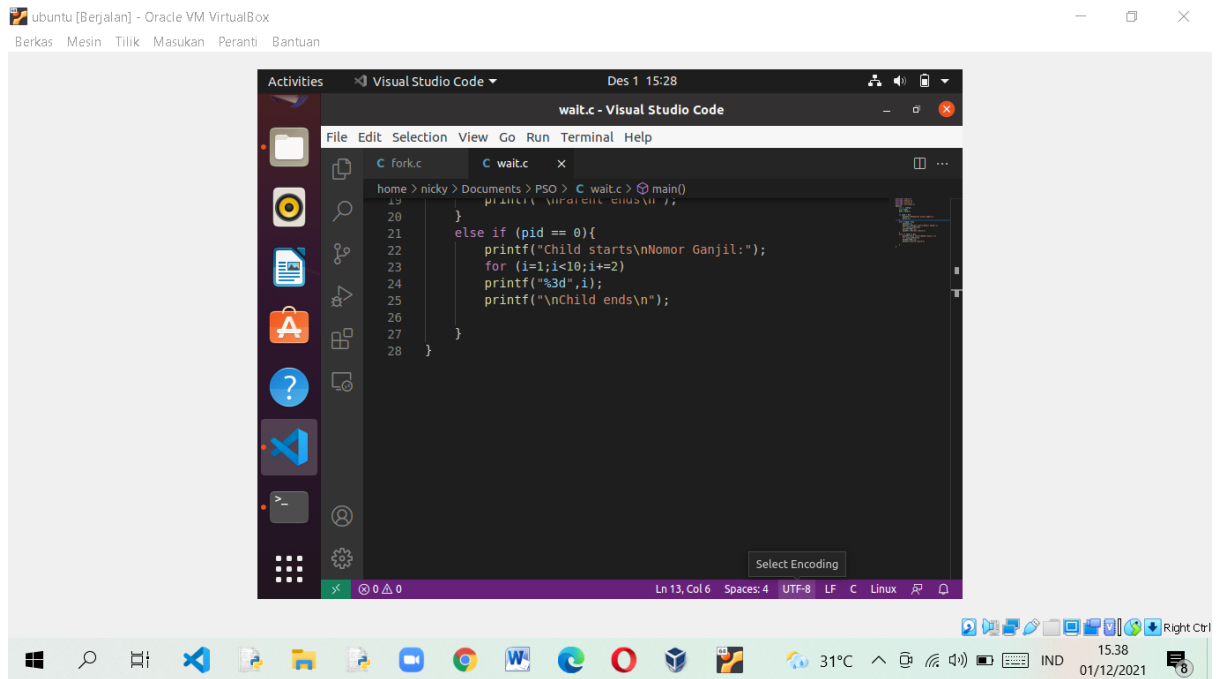
```
20 }
21 else{
22     printf("Parent process:");
23     printf("\nProces id is %d", getpid());
24     printf("\nValue of x is %d", x);
25     printf("\nProcess id of shell is %d\n", getpid());
26 }
27
28
29
30
```

The status bar at the bottom indicates the cursor is at line 7, column 15.

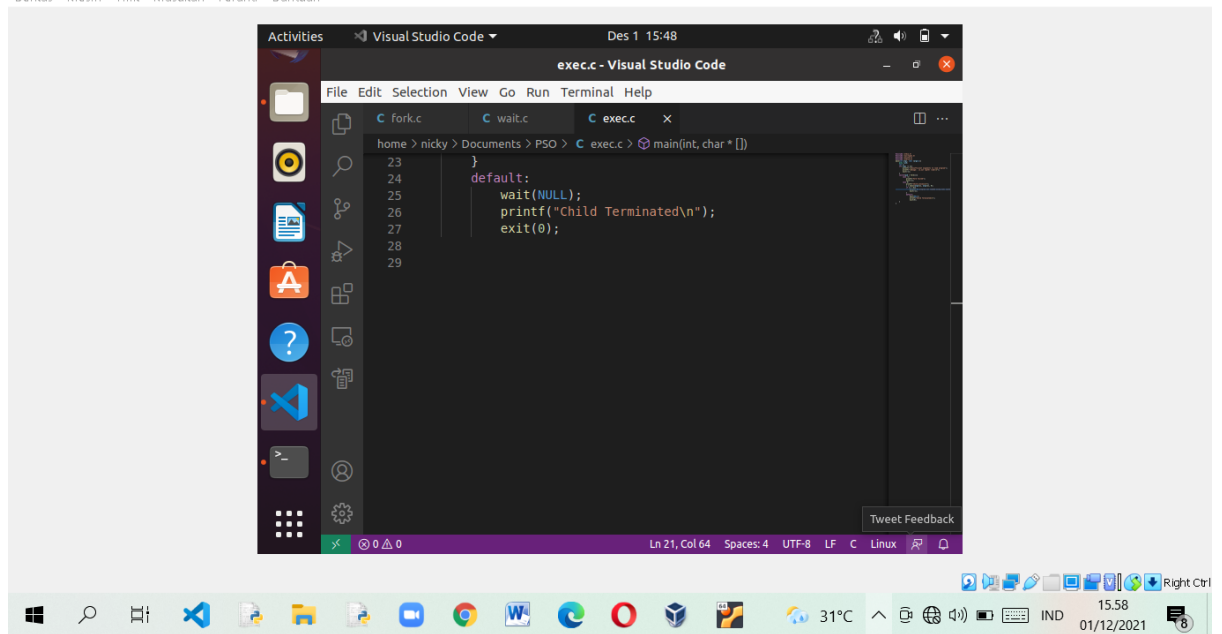
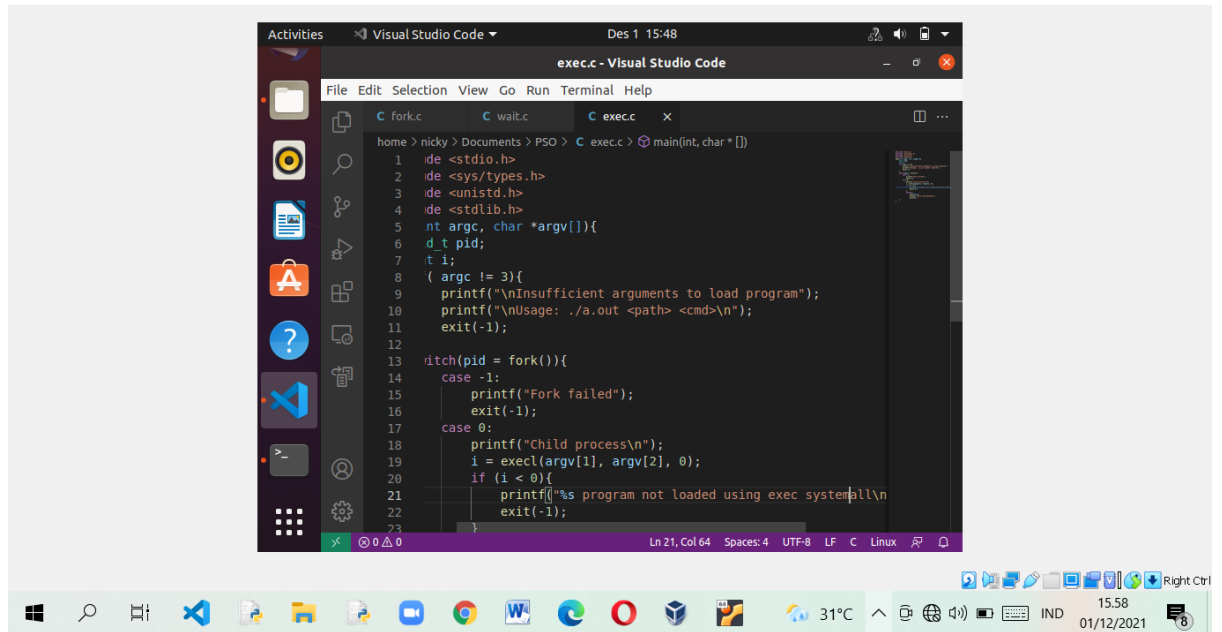
```
nicky@nicky-VirtualBox: ~/Documents/PSO
nicky@nicky-VirtualBox:~$ cd ~/Documents/PSO
nicky@nicky-VirtualBox:~/Documents/PSO$ gcc 'fork.c'
fork.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
6 | main(){
  | ~~~~~
nicky@nicky-VirtualBox:~/Documents/PSO$ ./a.out
Parent process:
Process id is 10327
Value of x is 6
Process id of shell is 10327
nicky@nicky-VirtualBox:~/Documents/PSO$ Child process:
Process id is 10328
Value of x is 6
Process id of parent is 10328
```

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

```
home > nicky > Documents > PSO > C wait.c > main()
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 main(){
6     int i, status;
7     pid_t pid;
8     pid = fork();
9
10    if (pid < 0){
11        printf("\nPembuatan proses gagal");
12        exit(-1);
13    }
14    else if (pid > 0){
15        wait(NULL);
16        printf("\nParent starts\nNomor Genap:");
17        for (i=2; i<=10; i+=2)
18            printf("%3d", i);
19        printf("\nParent ends\n");
20    }
21    else if (pid == 0){
22        printf("Child starts\nNomor Ganjil:");
23        for (i=1; i<=10; i+=2)
```



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



ubuntu [Berjalan] - Oracle VM VirtualBox

Berkas Mesin Tilik Masukan Peranti Bantuan

```
nick@nick-VirtualBox: ~/Documents/PSO
nick@nick-VirtualBox:~/Documents/PSO$ 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:19:13: warning: missing sentinel in function call [-Wformat=]
19 |     i = execl(argv[1], argv[2], 0);
    |           ^
exec.c:25:17: warning: implicit declaration of function 'wait' [-Wimplicit-function-declaration]
25 |     wait(NULL);
    |     ~~~~~
nick@nick-VirtualBox:~/Documents/PSO$ ./a.out
Insufficient arguments to load program
Usage: ./a.out <path> <cmd>
nick@nick-VirtualBox:~/Documents/PSO$ ./a.out /bin/ls ls
Child process
a.out exec.c fork.c wait.c
Child Terminated
nick@nick-VirtualBox:~/Documents/PSO$
```

Des 1 15:48

31°C 15.58 01/12/2021

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

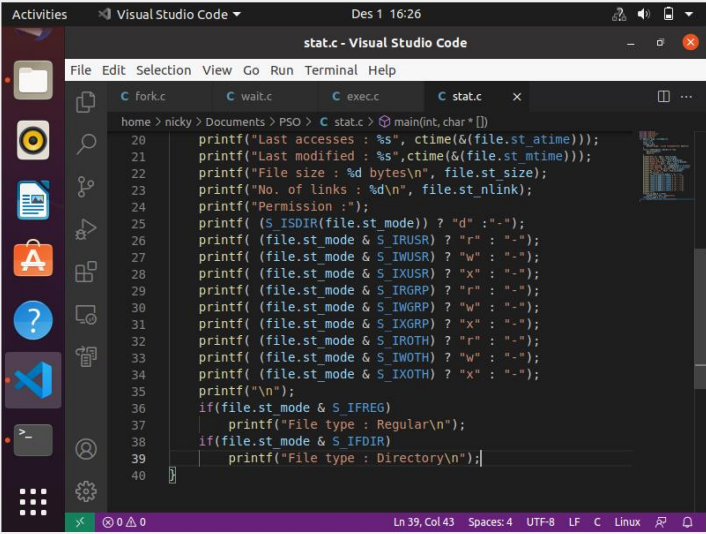
ubuntu [Berjalan] - Oracle VM VirtualBox

Berkas Mesin Tilik Masukan Peranti Bantuan

```
stat.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
C fork.c C wait.c C exec.c C stat.c x
home > nicky > Documents > PSO > C stat.c > main(int, char *[])
1 #include <stdio.h>
2 #include <sys/stat.h>
3 #include <stdlib.h>
4 #include <time.h>
5 int main(int argc, char*argv[])
6 {
7     struct stat
8     file; int n;
9     if(argc !=2){
10         printf("Usage: ./a.out <filenema>\n"); exit(-1);
11     }
12     if((n = stat(argv[1], &file)) == -1){
13         perror(argv[1]);
14         exit(-1);
15     }
16     printf("User id : %d\n", file.st_uid);
17     printf("Groub id : %d\n", file.st_gid);
18     printf("Block size : %d\n", file.st_blksize);
19     printf("Blocks allocated : %d\n", file.st_blocks);
20     printf("inode no. : %d\n", file.st_ino);
21     printf("Last accesses : %s", ctime(&(file.st_atime)));
22     printf("Last modified : %s", ctime(&(file.st_mtime)));
23     printf("File size : %d bytes\n", file.st_size);
24     printf("No. of links : %d\n", file.st_nlink);
25 }
```

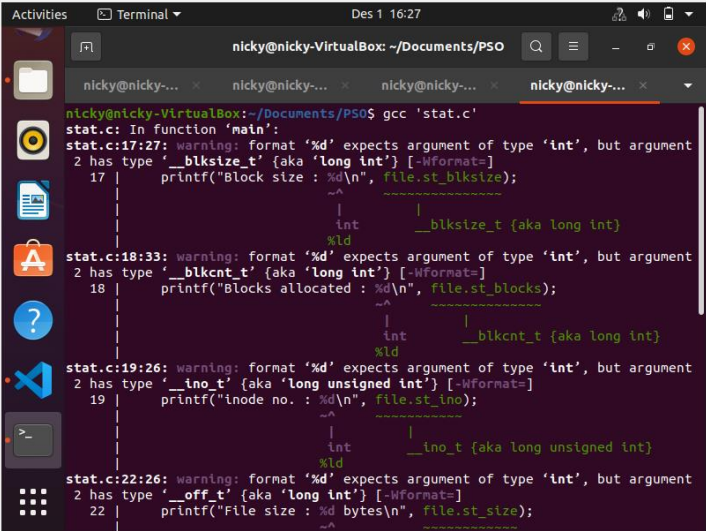
Des 1 16:26

31°C 16.36 01/12/2021



The screenshot shows the Visual Studio Code editor with the file `stat.c` open. The code is a C program that uses the `stat` system call to retrieve file status information. It prints out various details including last access and modification times, file size, number of links, permissions, and file type (regular file or directory). The code is written in C and uses `printf` for output.

```
home > nicky > Documents > PSO > C stat.c > main(int, char*[])
20 printf("Last accesses : %s", ctime(&(file.st_atime)));
21 printf("Last modified : %s", ctime(&(file.st_mtime)));
22 printf("File size : %d bytes\n", file.st_size);
23 printf("No. of links : %d\n", file.st_nlink);
24 printf("Permission :");
25 printf(" (%s_ISDIR(file.st_mode)) ? 'd' : '-');
26 printf(" (file.st_mode & S_IRUSR) ? 'r' : '-');
27 printf(" (file.st_mode & S_IWUSR) ? 'w' : '-');
28 printf(" (file.st_mode & S_IXUSR) ? 'x' : '-');
29 printf(" (file.st_mode & S_IRGRP) ? 'r' : '-');
30 printf(" (file.st_mode & S_IWGRP) ? 'w' : '-');
31 printf(" (file.st_mode & S_IXGRP) ? 'x' : '-');
32 printf(" (file.st_mode & S_IROTH) ? 'r' : '-');
33 printf(" (file.st_mode & S_IWOTH) ? 'w' : '-');
34 printf(" (file.st_mode & S_IXOTH) ? 'x' : '-');
35 printf("\n");
36 if(file.st_mode & S_IFREG)
37     printf("File type : Regular\n");
38 if(file.st_mode & S_IFDIR)
39     printf("File type : Directory\n");
40
```



The screenshot shows a terminal window where the `stat.c` file is being compiled using the `gcc` compiler. The output shows several warnings related to format specifiers and argument types. The warnings indicate that the format specifiers `%d` are being used with arguments of type `long int`, `long unsigned int`, and `long int`, which are not compatible with the `%d` format specifier.

```
nicky@nicky-VirtualBox: ~/Documents/PSO
nicky@nicky-VirtualBox:~/Documents/PSO$ gcc 'stat.c'
stat.c: In function 'main':
stat.c:17:27: warning: format '%d' expects argument of type 'int', but argument
2 has type '__blksize_t' {aka 'long int'} [-Wformat=]
17 | printf("Block size : %d\n", file.st_blksize);
   |                          ^
   |                          |
   |                          int      __blksize_t {aka long int}
   |                          %ld
stat.c:18:33: warning: format '%d' expects argument of type 'int', but argument
2 has type '__blkcnt_t' {aka 'long int'} [-Wformat=]
18 | printf("Blocks allocated : %d\n", file.st_blocks);
   |                          ^
   |                          |
   |                          int      __blkcnt_t {aka long int}
   |                          %ld
stat.c:19:26: warning: format '%d' expects argument of type 'int', but argument
2 has type '__ino_t' {aka 'long unsigned int'} [-Wformat=]
19 | printf("inode no. : %d\n", file.st_ino);
   |                          ^
   |                          |
   |                          int      __ino_t {aka long unsigned int}
   |                          %ld
stat.c:22:26: warning: format '%d' expects argument of type 'int', but argument
2 has type '__off_t' {aka 'long int'} [-Wformat=]
22 | printf("File size : %d bytes\n", file.st_size);
   |                          ^
   |                          |
   |                          int      __off_t {aka long int}
   |                          %ld
```


ubuntu [Berjalan] - Oracle VM VirtualBox

Berkas Mesin Tilik Masukan Peranti Bantuan

```
nick@nick-VirtualBox: ~/Documents/PSO
nick@nick-VirtualBox:~/Documents/PSO$ ./a.out
stat.c:23:29: warning: format '%d' expects argument of type 'int', but argument 2 has type '__nlink_t' {aka 'long unsigned int'} [-Wformat=]
23 | printf("No. of links : %d\n", file.st_nlink);
    |                        ^
    |                        |
    |                        int
    |                        |
    |                        __nlink_t {aka long unsigned int}

nick@nick-VirtualBox:~/Documents/PSO$ ./a.out
Usage: ./a.out <filenema>
nick@nick-VirtualBox:~/Documents/PSO$ ./a.out fork.c
User id : 1000
Group id : 1000
Block size : 4096
Blocks allocated : 8
Inode no. : 799394
Last accesses : Wed Dec 1 15:10:25 2021
Last modified : Wed Dec 1 15:09:37 2021
File size : 640 bytes
No. of links : 1
Permission : -rw-rw-r--
File type : Regular
nick@nick-VirtualBox:~/Documents/PSO$
```

31°C 16.39 01/12/2021

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

ubuntu [Berjalan] - Oracle VM VirtualBox

Berkas Mesin Tilik Masukan Peranti Bantuan

```
dirlist.c - Visual Studio Code
File Edit Selection View Go Run Terminal Help
fork.c wait.c exec.c stat.c dirlist.c
home > nicky > Desktop > C dirlist.c > ...
1 #include <stdio.h>
2 #include <dirent.h>
3 #include <stdlib.h>
4 main(int argc, char *argv[]){
5     struct dirent *dptr;
6     DIR *dname;
7
8     if(argc !=2){
9         printf("Usage: ./a.out <dirname>\n");
10        exit(-1);
11    }
12    if ((dname = opendir(argv[1])) == NULL){
13        perror(argv[1]);
14        exit(-1);
15    }
16    while(dptr=readdir(dname))
17        printf("%s\n", dptr->d_name);
18
19    closedir(dname);
20
21
22
Ln 21, Col 1 Spaces: 4 UTF-8 LF C Linux
```

31°C 17.01 01/12/2021


```

Activities Terminal Des 1 16:55
nicky@nicky-VirtualBox: ~/Desktop
nicky@nicky-VirtualBox: ~/Desktop
nicky@nicky-VirtualBox:~/Desktop$ gcc 'dirlist.c'
dirlist.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
4 | main(int argc, char *argv[]){
  | ^~~~~~
nicky@nicky-VirtualBox:~/Desktop$ ./a.out P50
exec.c
dirlist
fork.c
dirlist.c
.
a.out
stat.c
..
wait.c
nicky@nicky-VirtualBox:~/Desktop$

```