

Nama : Rohmad Khoirudin
NIM : L200180101
Kelas : C

MODUL 8

Stat.c

```

root@convy22nd-Lenovo-ideapad-100-14IBD:/home/convy_22nd/Documents# clear
root@convy22nd-Lenovo-ideapad-100-14IBD:/home/convy_22nd/Documents# gcc stat.c
stat.c: In function 'main':
stat.c:19:26: warning: format '%d' expects argument of type 'int', but argument 2 has type '__blksize_t {aka long int}' [-Wformat=]
   printf("Block size: %d\n", file.st_blksize);
                        ^
                        %ld
stat.c:20:31: warning: format '%d' expects argument of type 'int', but argument 2 has type '__blkcnt_t {aka long int}' [-Wformat=]
   printf("Block allocated: %d\n", file.st_blocks);
                               ^
                               %ld
stat.c:21:26: 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:24:26: 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:25:29: 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
root@convy22nd-Lenovo-ideapad-100-14IBD:/home/convy_22nd/Documents# ./a.out stat.c
User id : 1000
Group id: 1000
Block size: 4096
Block allocated: 8
Inode no. : 941296
Last accessed : Sat Nov 30 22:30:01 2019
Last modified : Tue Nov 19 13:38:19 2019
File size : 1519 bytes
No. of links : 1
Permission : -rw-r--r--
File type : Regular
root@convy22nd-Lenovo-ideapad-100-14IBD:/home/convy_22nd/Documents#

```

Source code stat.c

The screenshot shows a terminal window with the nano text editor open, editing a file named `stat.c`. The code is a C program that uses the `stat` system call to display file statistics for a given file. The program includes headers for `stdio.h`, `stdlib.h`, `time.h`, and `sys/stat.h`. It defines a `main` function that takes command-line arguments, checks for the correct number of arguments, and then calls `stat` on the specified file. If successful, it prints various file attributes such as user ID, group ID, block size, allocated blocks, inode number, last accessed/modified times, file size, number of links, and permissions. It also prints the file type (regular, directory, etc.) and the owner/group names.

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <sys/stat.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("Block 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("Permission : ");
    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" : "-");
}

```

The terminal window title is "Terminal - root@convy22nd-lenovo-ideapad-100-1418D: /home/convy_22nd/Documents". The nano editor's status bar at the bottom shows "GNU nano 2.9.3 stat.c". The bottom status bar of the terminal window displays various keyboard shortcuts for navigation and editing.

```
GNU nano 2.9.3 stat.c

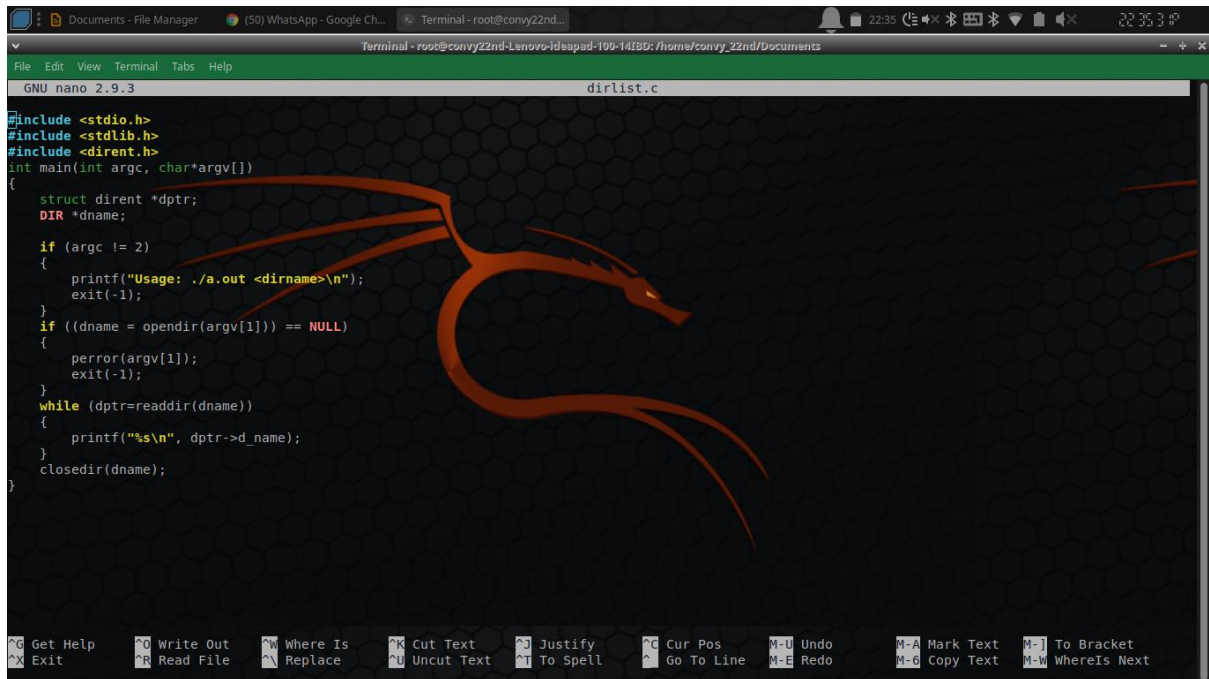
printf("User id : %d\n", file.st_uid);
printf("Group id: %d\n", file.st_gid);
printf("Block size: %d\n", file.st_blksize);
printf("Block allocated: %d\n", file.st_blocks);
printf("Inode no. : %d\n", file.st_ino);
printf("Last accessed : %s", ctime(&(file.st_atime)));
printf("Last modified : %s", ctime(&(file.st_mtime)));
printf("File size : %d bytes\n", file.st_size);
printf("No. of links : %d\n", file.st_nlink);
printf("Permission : ");
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");
}
}
```

Dirlist.c

```
convy_22nd@convy22nd-Lenovo-ideapad-100-141BD:~/Documents$ clear
convy_22nd@convy22nd-Lenovo-ideapad-100-141BD:~/Documents$ sudo su
[sudo] password for convy_22nd:

root@convy22nd-Lenovo-ideapad-100-141BD:/home/convy_22nd/Documents# gcc dirlist.c
root@convy22nd-Lenovo-ideapad-100-141BD:/home/convy_22nd/Documents# ./a.out .
stat.c
stat.c.save
..
Problem Solving LO Makrab
a.out
exploit-linuxsec-org-tutorial-sql-injection-manual-.pdf
diskrit
tugasTambahan
kaliBlackMain.png
jcalendar-1.4.jar
dirlist.c
latian uts
Untitled 1.odt
Project PBO
.
root@convy22nd-Lenovo-ideapad-100-141BD:/home/convy_22nd/Documents#
```

Source code dirlist.c



The screenshot shows a terminal window with the nano text editor open, editing a file named `dirlist.c`. The code is written in C and uses the `dirent` library to list directory contents. A large, stylized orange dragon logo is visible in the background of the editor. The terminal window has a title bar that reads "Terminal - root@convy22nd-Lenovo-ideapad-100-14IBD: /home/convy_22nd/Documents". The nano editor's status bar at the bottom shows "GNU nano 2.9.3". The code is as follows:

```
#include <stdio.h>
#include <stdlib.h>
#include <dirent.h>
int 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);
}
```

The bottom of the terminal window displays a row of nano editor shortcuts: Get Help, Exit, Write Out, Read File, Where Is, Replace, Cut Text, Uncut Text, Justify, To Spell, Cur. Pos, Go To Line, Undo, Redo, Mark Text, Copy Text, To Bracket, and WhereIs Next.