

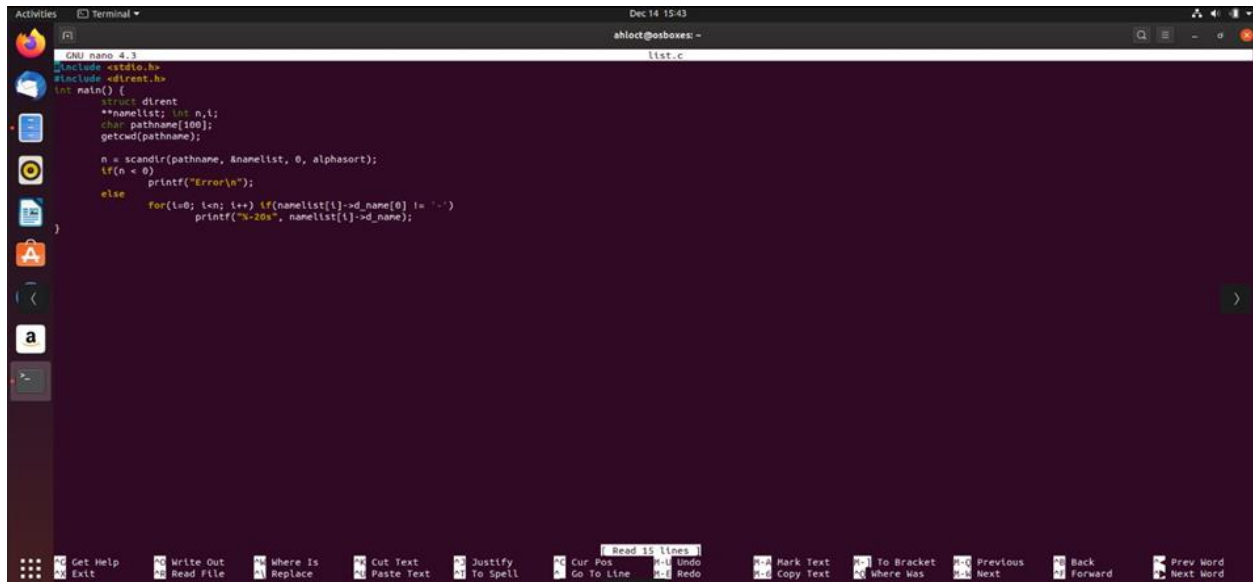
NAMA : HANAN ASKARIM

NIM : L200180170

KELAS : D

MODUL 10

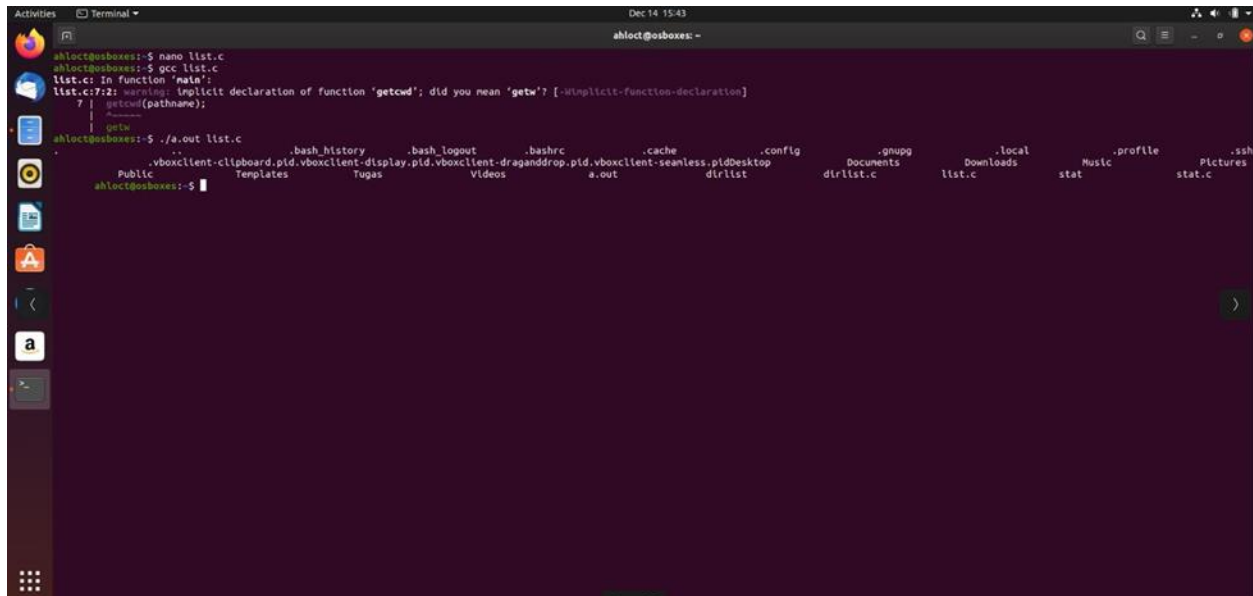
## 1. List



```
GNU nano 4.3 list.c
#include <stdio.h>
#include <dirent.h>

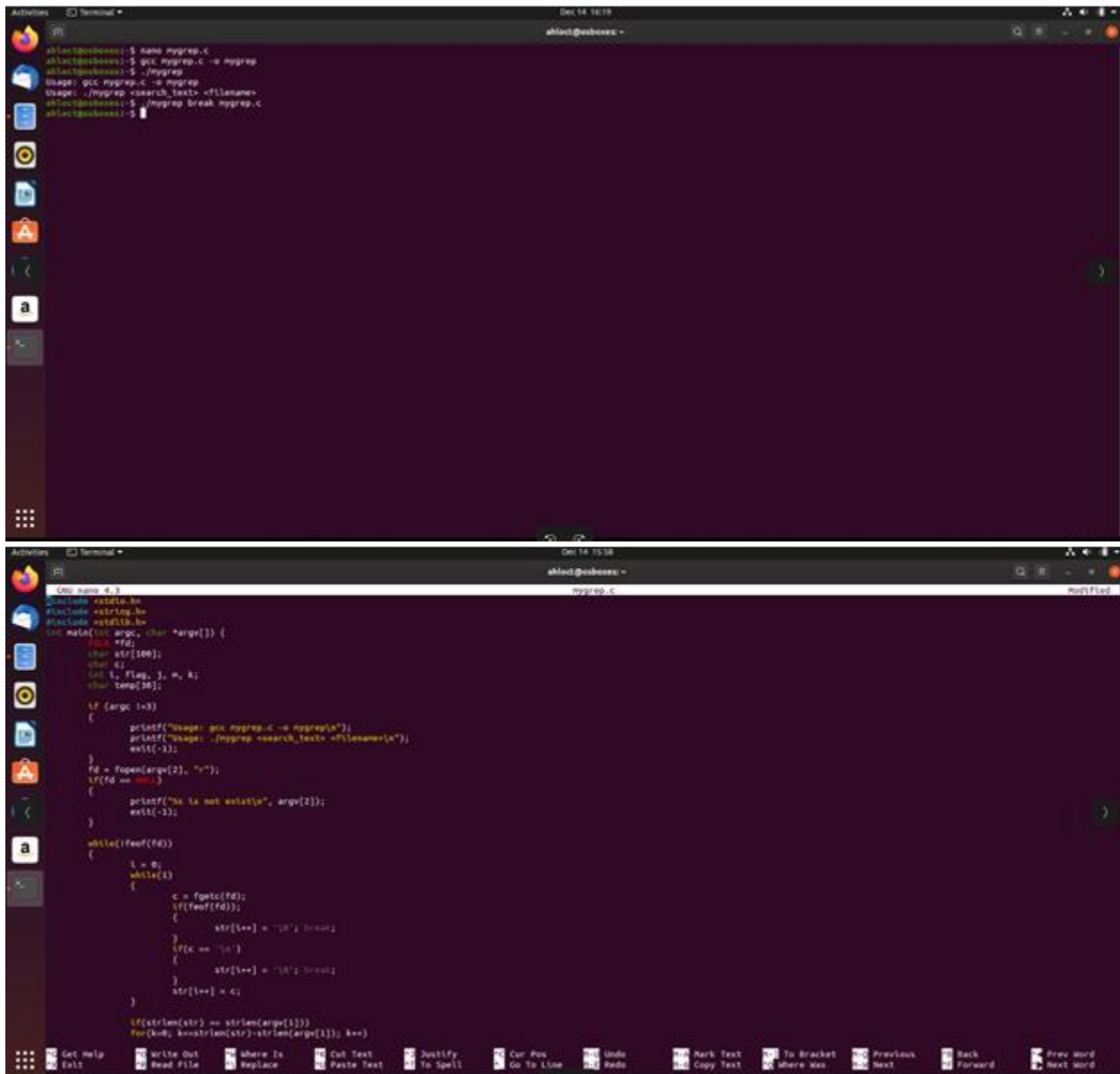
int main() {
    struct dirent
    **namelist; int n,i;
    char pathname[100];
    getcwd(pathname);

    n = scandir(pathname, &namelist, 0, alphasort);
    if(n < 0)
        printf("Error\n");
    else
        for(i=0; i<n; i++) if(namelist[i]->d_name[0] != '.')
            printf("%-20s", namelist[i]->d_name);
}
```



```
ahloct@osboxes:~$ nano list.c
ahloct@osboxes:~$ gcc list.c
list.c: In function 'main':
list.c:7:2: warning: implicit declaration of function 'getcwd'; did you mean 'getw'? [-Wimplicit-function-declaration]
7 |   getcwd(pathname);
  |   ^~~~~~
ahloct@osboxes:~$ ./a.out list.c
.bash_history      .bash_logout      .bashrc           .cache            .config           .gnupg            .local            .profile          .ssh
.vboxclient-clipboard.pid.vboxclient-display.pid.vboxclient-draganddrop.pid.vboxclient-seamless.pid.Desktop.Documents        Downloads         Music             Pictures
Public             Templates         Tugus             Videos           a.out             dirlist.c         list.c            stat              stat.c
ahloct@osboxes:~$
```

## 2. Mygrep



```
ahlect@ubuntu:~$ nano mygrep.c
ahlect@ubuntu:~$ gcc mygrep.c -o mygrep
ahlect@ubuntu:~$ ./mygrep
usage: gcc mygrep.c -o mygrep
ahlect@ubuntu:~$ ./mygrep -search_text <filename>
ahlect@ubuntu:~$ ./mygrep break mygrep.c
ahlect@ubuntu:~$
```

```

1  #include <stdio.h>
2  #include <string.h>
3  #include <stdlib.h>
4  int main(int argc, char *argv[]) {
5      FILE *fd;
6      char str[100];
7      char *k;
8      int i, flag, j, n, k;
9      char temp[10];
10
11     if (argc != 3)
12     {
13         printf("usage: gcc mygrep.c -o mygrep\n");
14         printf("usage: ./mygrep -search_text <filename>\n");
15         exit(-1);
16     }
17     fd = fopen(argv[2], "r");
18     if (fd == NULL)
19     {
20         printf("file is not exist\n", argv[2]);
21         exit(-1);
22     }
23     while (!feof(fd))
24     {
25         i = 0;
26         while (1)
27         {
28             c = fgetc(fd);
29             if (feof(fd))
30             {
31                 str[i++] = '\0'; break;
32             }
33             if (c == '\n')
34             {
35                 str[i++] = '\0'; break;
36             }
37             str[i++] = c;
38         }
39         if (strcmp(str, argv[1]) == 0)
40             printf("%s\n", str);
41     }
42     fclose(fd);
43 }
```

```
Activities Terminal Dec 14 15:58
ahloct@osboxes: ~
CND nano 4.3 mygrep.c Modified

printf("Usage: ./mygrep <search_text> <filenames>\n");
exit(-1);

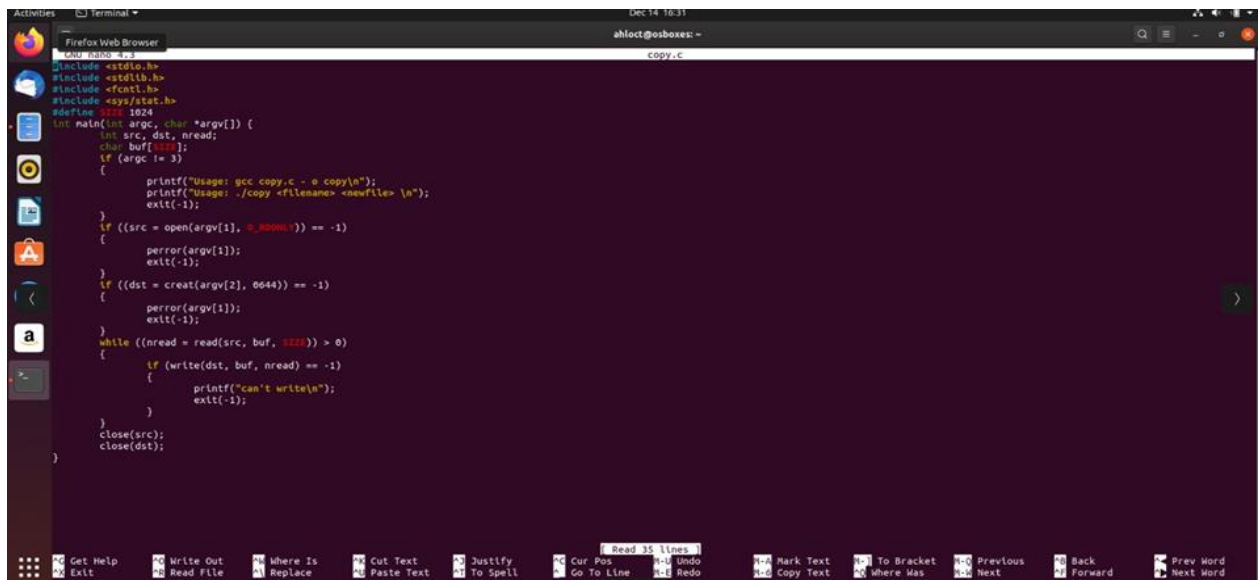
fd = fopen(argv[2], "r");
if(fd == NULL)
{
    printf("File is not exist\n", argv[2]);
    exit(-1);
}

while(!feof(fd))
{
    l = 0;
    while(1)
    {
        c = fgetc(fd);
        if(feof(fd))
        {
            str[l++] = '\0'; break;
        }
        if(c == '\n')
        {
            str[l++] = '\0'; break;
        }
        str[l++] = c;
    }

    if(strlen(str) == strlen(argv[1]))
    {
        for(k=0; k<strlen(str)-strlen(argv[1]); k++)
        {
            for(m=0; m<strlen(argv[1]);m++)
            {
                temp[m] = str[k+m];
                if(strcmp(temp, argv[1]) == 0)
                {
                    printf("%s\n", str);
                    break;
                }
            }
        }
    }
}

1
Get Help Write Out Where Is Cut Text Justify Cur Pos Undo Mark Text To Bracket Previous Back Prev Word
Exit Read File Replace Paste Text To Spell Go To Line Redo Copy Text Where Was Next Next Forward Next Word
```

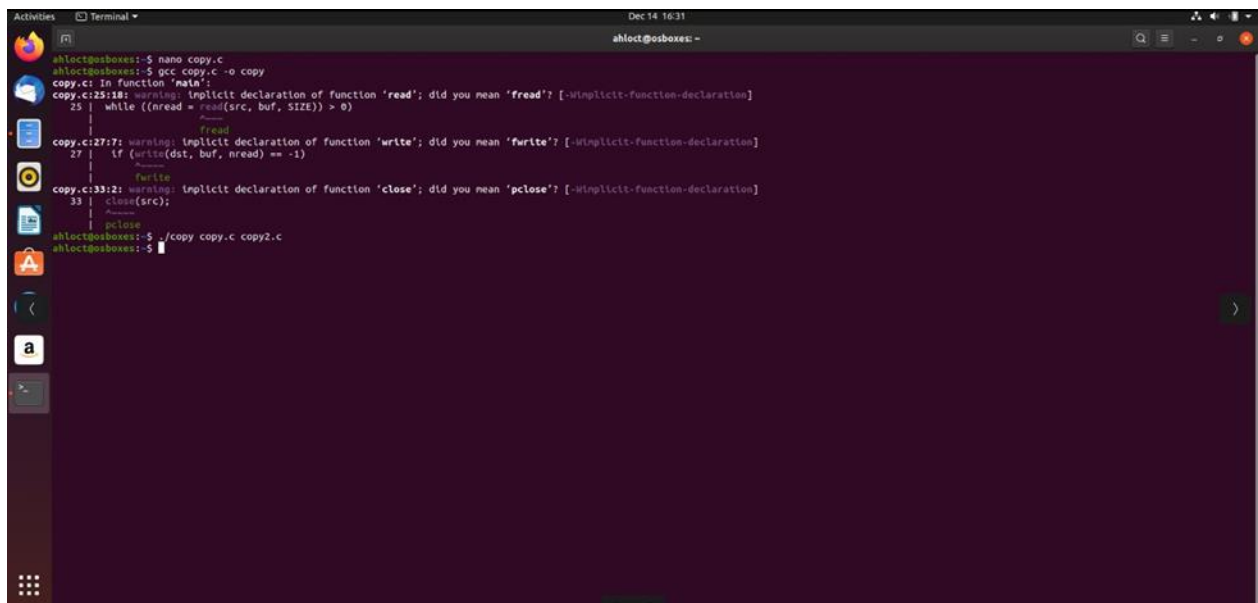
### 3. Copy



The screenshot shows a terminal window with the file 'copy.c' open in a text editor. The code is a C program that copies the contents of one file to another. It includes standard headers like `<stdio.h>`, `<stdlib.h>`, `<fcntl.h>`, and `<sys/stat.h>`. It defines a buffer size of 1024 and uses `main` to handle command-line arguments. The program opens the source file, creates the destination file, and then reads from the source and writes to the destination in a loop until the end of the file is reached. Error handling is present for file opening and writing failures.

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#define SIZE 1024

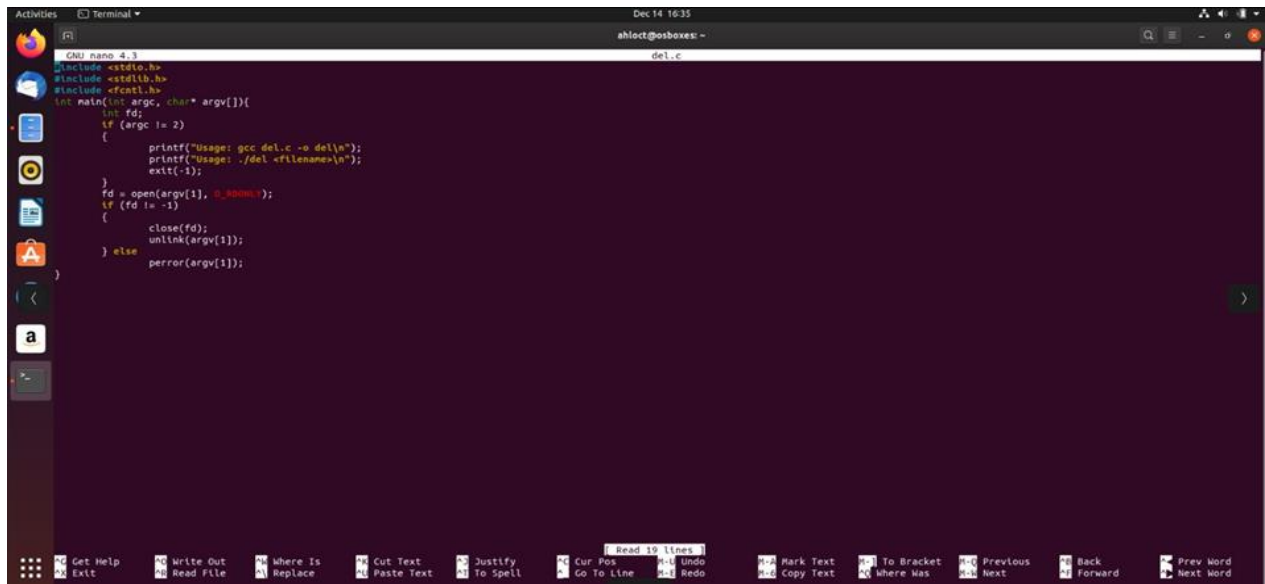
int main(int argc, char *argv[]) {
    int src, dst, nread;
    char buf[SIZE];
    if (argc != 3) {
        printf("Usage: gcc copy.c -o copy\n");
        printf("Usage: ./copy <filename> <newfile> \n");
        exit(-1);
    }
    if ((src = open(argv[1], O_RDONLY)) == -1) {
        perror(argv[1]);
        exit(-1);
    }
    if ((dst = creat(argv[2], 0644)) == -1) {
        perror(argv[2]);
        exit(-1);
    }
    while ((nread = read(src, buf, SIZE)) > 0) {
        if (write(dst, buf, nread) == -1) {
            printf("can't write\n");
            exit(-1);
        }
    }
    close(src);
    close(dst);
}
```



The screenshot shows the terminal output after compiling and running the program. The user runs `gcc copy.c` to compile the program into an executable named `copy`. Then, they run `./copy copy.c copy2.c` to copy the source file into a new file. The output shows several warnings from GCC about implicit function declarations for `read`, `write`, and `close`, which are resolved by including the appropriate headers. The program runs successfully without any errors.

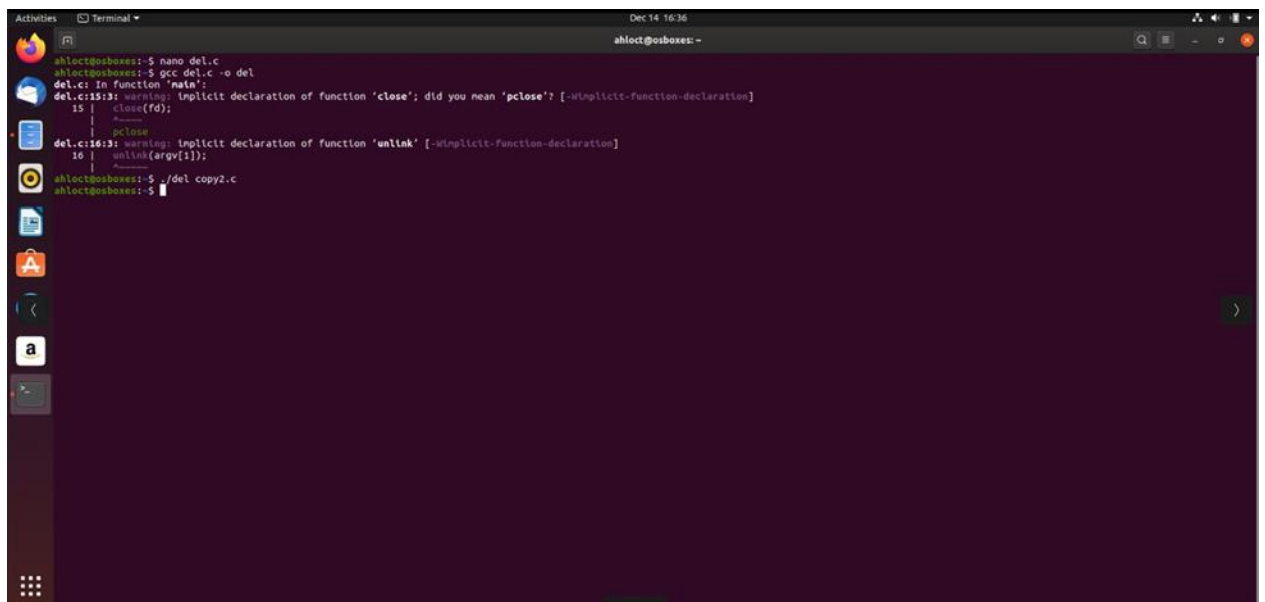
```
ahloct@osboxes:~$ gcc copy.c
ahloct@osboxes:~$ gcc copy.c -o copy
copy.c: In function 'main':
copy.c:25:18: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
25 | while ((nread = read(src, buf, SIZE)) > 0)
    |                  ^~~~~
    |                  fread
copy.c:27:7: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
27 | if (write(dst, buf, nread) == -1)
    |     ^~~~~
    |     fwrite
copy.c:33:12: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
33 | close(src);
    |     ^~~~~
    |     pclose
ahloct@osboxes:~$ ./copy copy.c copy2.c
ahloct@osboxes:~$
```

## 4. Del



```
GNU nano 4.3 del.c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>

int main(int argc, char* argv[]){
    int fd;
    if (argc != 2)
    {
        printf("Usage: gcc del.c -o del\n");
        printf("Usage: ./del <filename>\n");
        exit(-1);
    }
    fd = open(argv[1], O_RDONLY);
    if (fd != -1)
    {
        close(fd);
        unlink(argv[1]);
    } else
        perror(argv[1]);
}
```



```
ahloct@osboxes: ~$ nano del.c
ahloct@osboxes: ~$ gcc del.c -o del
del.c: In function 'main':
del.c:15:13: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
15 |     close(fd);
    |     ^~~~~
del.c:16:13: warning: implicit declaration of function 'unlink' [-Wimplicit-function-declaration]
16 |     unlink(argv[1]);
    |     ^~~~~
ahloct@osboxes: ~$ ./del copy2.c
ahloct@osboxes: ~$
```