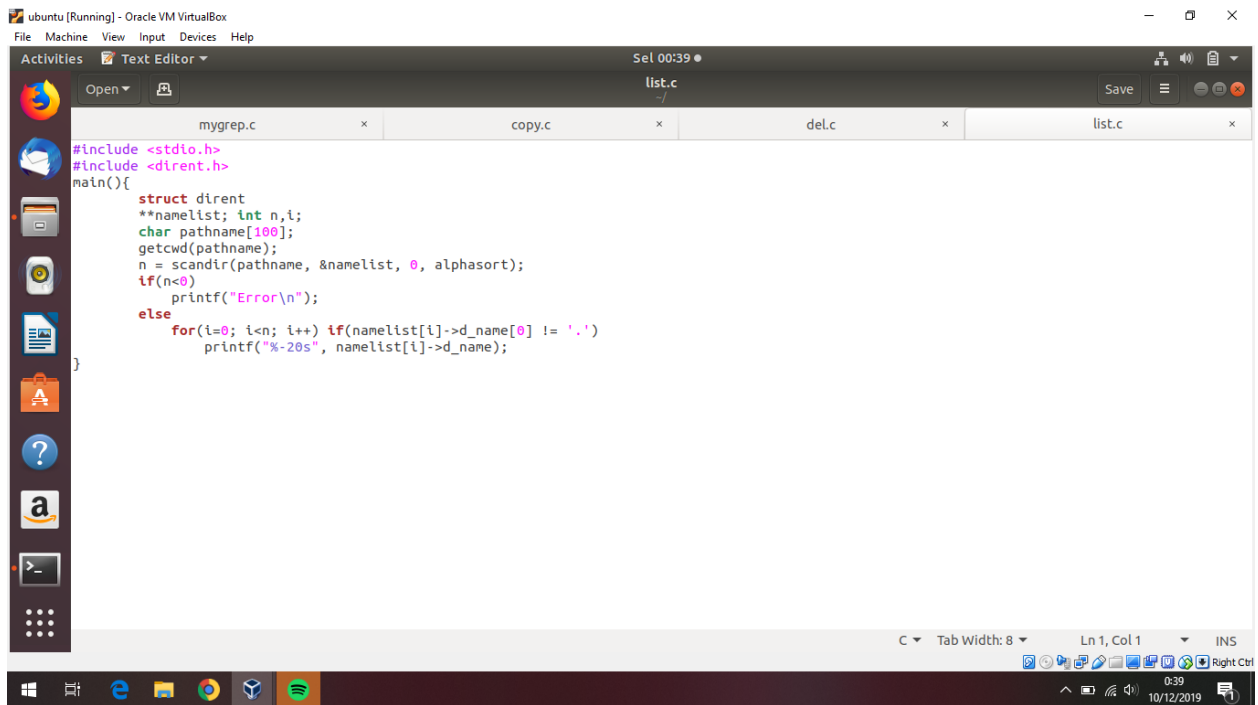


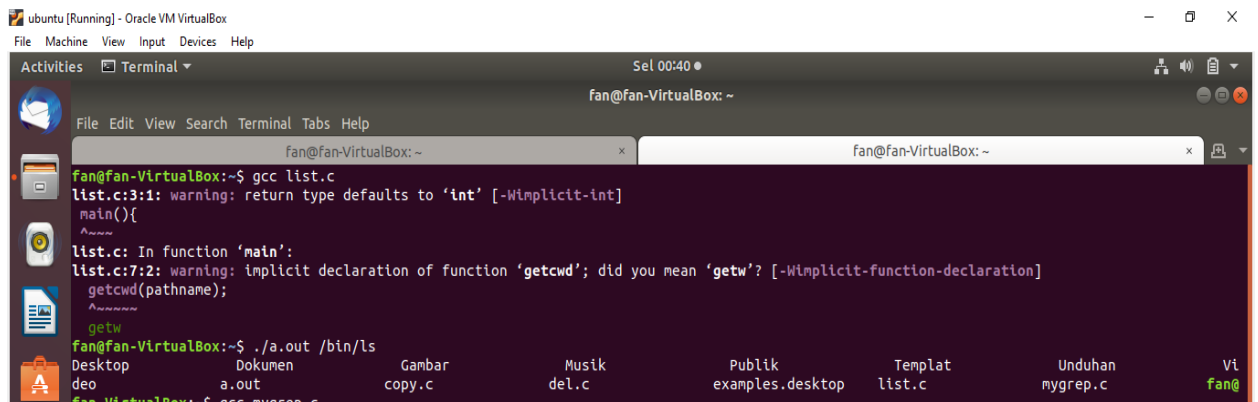
List.c



The screenshot shows a text editor window titled 'list.c' with the following code:

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

The editor interface includes a menu bar (File, Machine, View, Input, Devices, Help), a toolbar with icons for opening files, saving, and undo/redo, and a status bar at the bottom showing 'Ln 1, Col 1' and 'INS'.

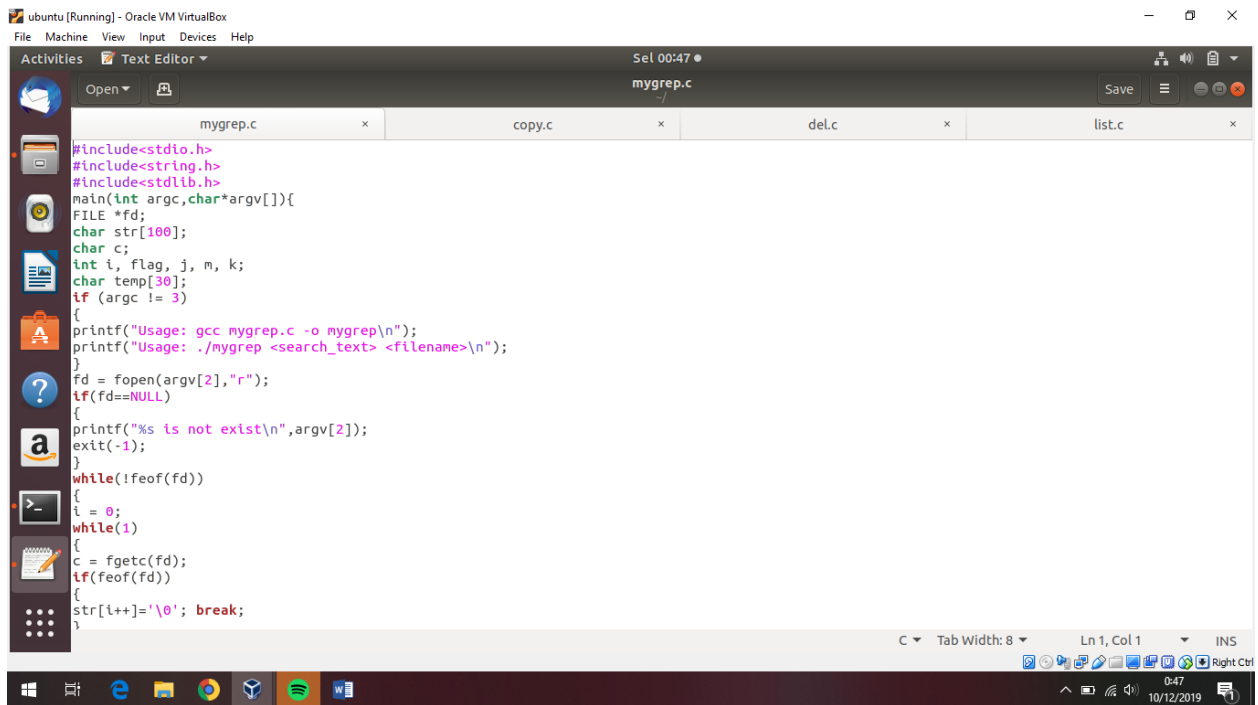


The screenshot shows a terminal window with the following output:

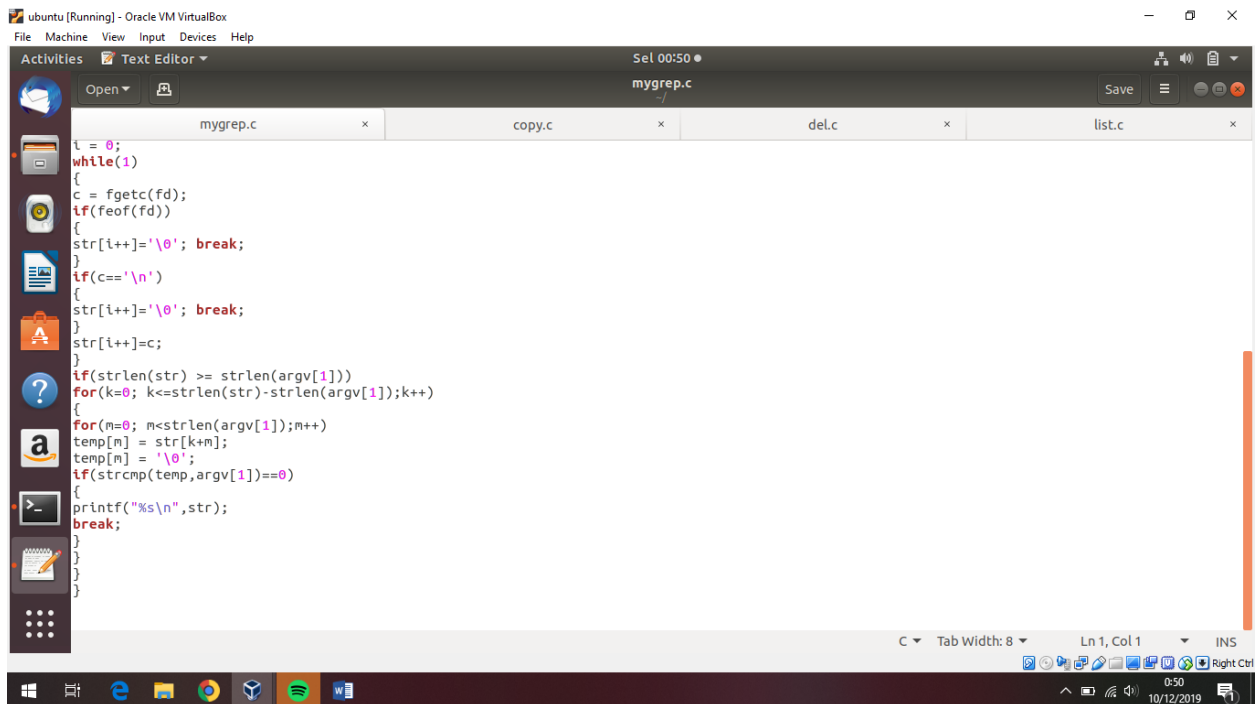
```
fan@fan-VirtualBox: ~
fan@fan-VirtualBox:~$ gcc list.c
list.c:3:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(){
^~~~~~
list.c: In function 'main':
list.c:7:2: warning: implicit declaration of function 'getcwd'; did you mean 'getw'? [-Wimplicit-function-declaration]
getcwd(pathname);
^~~~~~
getw
fan@fan-VirtualBox:~$ ./a.out /bin/ls
Desktop      Dokumen      Gambar      Musik      Publik      Templat      Unduhan      Vi
deo          a.out       copy.c      del.c      examples.desktop list.c       mygrep.c    fan@
```

The terminal window has a menu bar (File, Edit, View, Search, Terminal, Tabs, Help) and a status bar at the bottom showing the current directory and the user's name.

Mygrep.c

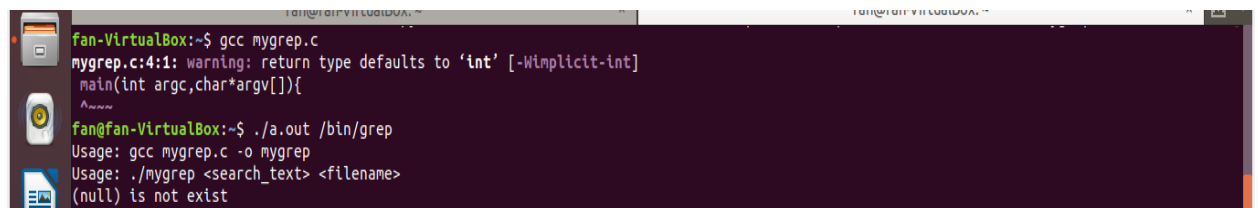


```
mygrep.c
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
main(int argc,char*argv[]){
FILE *fd;
char str[100];
char c;
int i, flag, j, m, k;
char temp[30];
if (argc != 3)
{
printf("Usage: gcc mygrep.c -o mygrep\n");
printf("Usage: ./mygrep <search_text> <filename>\n");
}
fd = fopen(argv[2],"r");
if(fd==NULL)
{
printf("%s is not exist\n",argv[2]);
exit(-1);
}
while(!feof(fd))
{
i = 0;
while(1)
{
c = fgetc(fd);
if(feof(fd))
{
str[i++]='\0'; break;
}
```



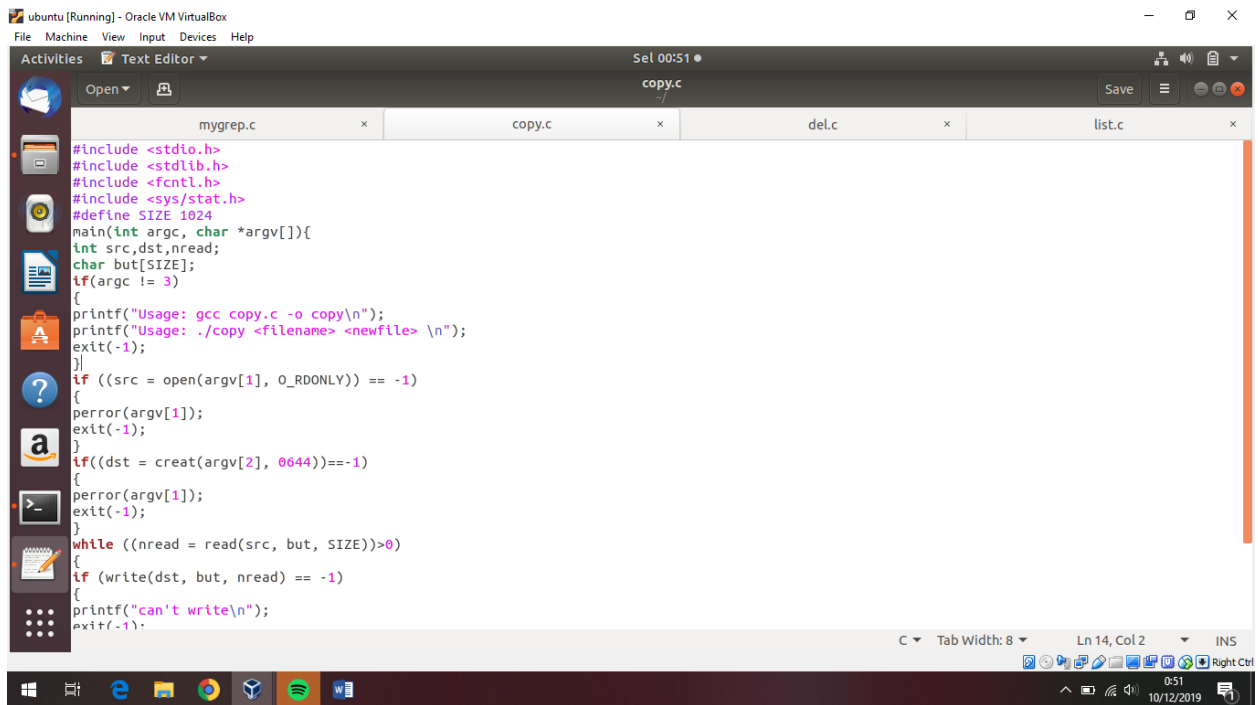
```

i = 0;
while(1)
{
c = fgetc(fd);
if(feof(fd))
{
str[i++]='\0'; break;
}
if(c=='\n')
{
str[i++]='\0'; break;
}
str[i++] = 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];
temp[m] = '\0';
if(strcmp(temp,argv[1])==0)
{
printf("%s\n",str);
break;
}
}
}
```



```
fan-VirtualBox:~$ gcc mygrep.c
mygrep.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc,char*argv[]){
^
fan@fan-VirtualBox:~$ ./a.out /bin/grep
Usage: gcc mygrep.c -o mygrep
Usage: ./mygrep <search_text> <filename>
(null) is not exist
```

Copy.c



ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

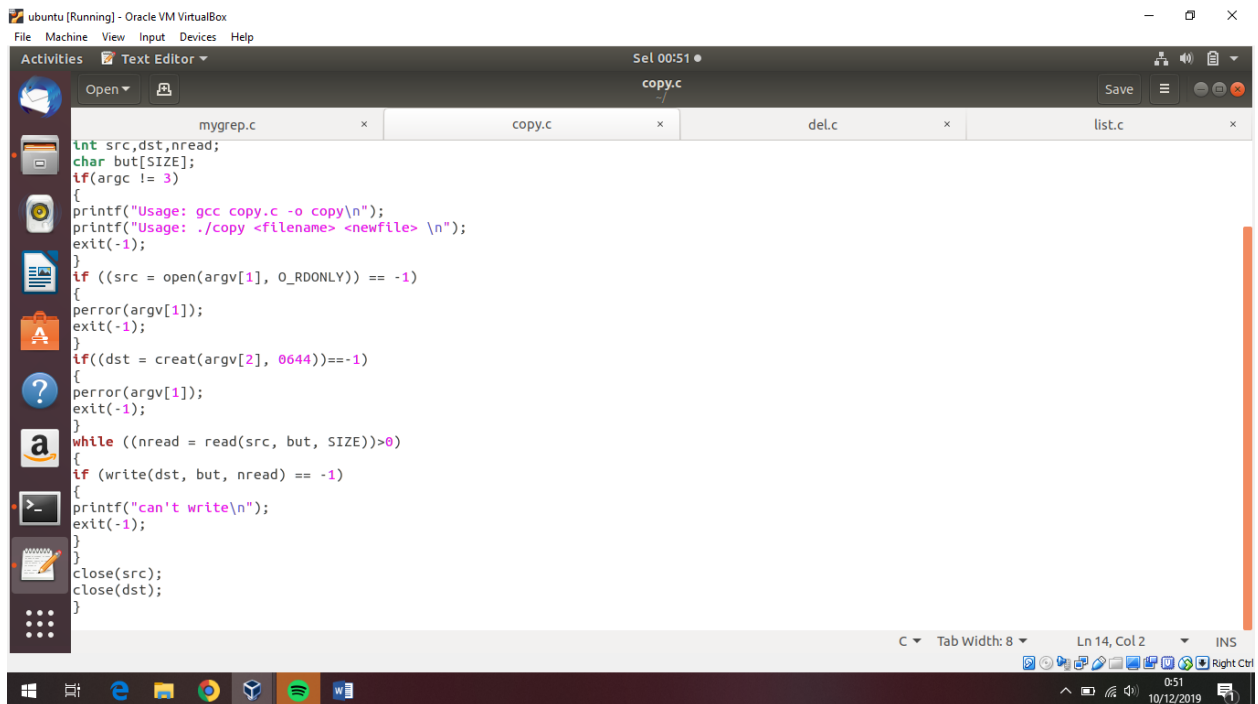
Activities Text Editor Sel 00:51

Open copy.c Save

mygrep.c x copy.c x del.c x list.c x

```
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#define SIZE 1024
main(int argc, char *argv[]){
    int src,dst,nread;
    char but[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[1]);
        exit(-1);
    }
    while ((nread = read(src, but, SIZE))>0)
    {
        if (write(dst, but, nread) == -1)
        {
            printf("can't write\n");
            exit(-1);
        }
    }
}
```

Tab Width: 8 Ln 14, Col 2 INS



ubuntu [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help

Activities Text Editor Sel 00:51

Open copy.c Save

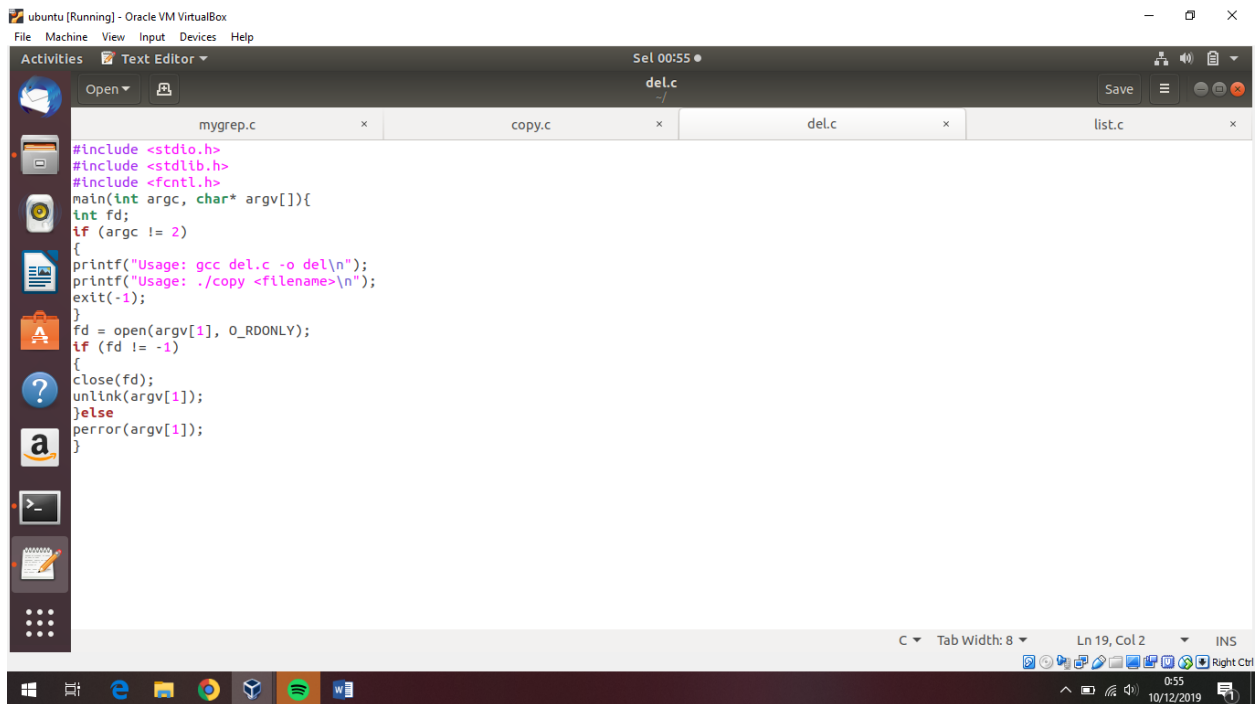
mygrep.c x copy.c x del.c x list.c x

```
int src,dst,nread;
char but[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[1]);
    exit(-1);
}
while ((nread = read(src, but, SIZE))>0)
{
    if (write(dst, but, nread) == -1)
    {
        printf("can't write\n");
        exit(-1);
    }
}
close(src);
close(dst);
}
```

Tab Width: 8 Ln 14, Col 2 INS

```
fan@fan-VirtualBox:~$ gcc copy.c
copy.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char *argv[]){
^
copy.c: In function 'main':
copy.c:25:17: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
while ((nread = read(src, but, SIZE))>0)
                ^read
copy.c:27:5: warning: implicit declaration of function 'write'; did you mean 'fwrite'? [-Wimplicit-function-declaration]
if (write(dst, but, nread) == -1)
    ^write
copy.c:33:1: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
close(src);
^close
pclose
fan@fan-VirtualBox:~$ ./a.out /bin/cp
Usage: gcc copy.c -o copy
Usage: ./copy <filename> <newfile>
```

Del.c



```
del.c
~/
Save
mygrep.c x copy.c x del.c x list.c x
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
main(int argc, char* argv[]){
    int fd;
    if (argc != 2)
    {
        printf("Usage: gcc del.c -o del\n");
        printf("Usage: ./copy <filename>\n");
        exit(-1);
    }
    fd = open(argv[1], O_RDONLY);
    if (fd != -1)
    {
        close(fd);
        unlink(argv[1]);
    }else
        perror(argv[1]);
}
```

```
fan@fan-VirtualBox:~$ gcc del.c
del.c:4:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char* argv[]){
^
del.c: In function 'main':
del.c:15:1: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
close(fd);
^close
del.c:16:1: warning: implicit declaration of function 'unlink'; did you mean 'unix'? [-Wimplicit-function-declaration]
unlink(argv[1]);
^unix
fan@fan-VirtualBox:~$ ./a.out /bin/rm
fan@fan-VirtualBox:~$
```