

**Nama : Iqbal Ramadhani**

**NIM : L200180155**

**Kelas : D**

**Praktikum SISTEM OPERASI**  
**Modul 10 (Simulasi Command)**

## 1. List

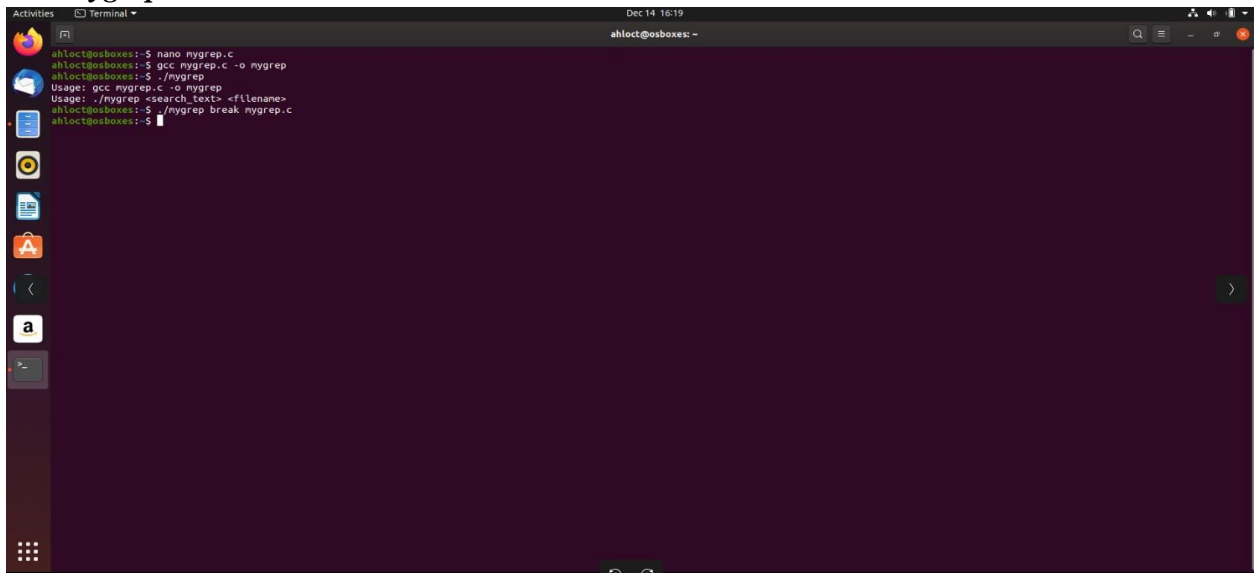
The first screenshot shows the compilation of 'list.c' using 'gcc'. The command 'ahloct@osboxes:~\$ gcc list.c' is entered, resulting in a warning about an implicit declaration of 'getcwd' and the creation of an executable 'a.out'. The command 'ahloct@osboxes:~\$ ./a.out list.c' is then entered, and the terminal displays a list of files and directories in the current directory, including '.bash\_history', '.bash\_logout', '.bashrc', '.cache', '.config', '.gnupg', '.local', '.profile', '.ssh', 'Public', 'Templates', 'Tugas', 'Videos', 'a.out', 'dirlist', 'dirlist.c', 'Downloads', 'list.c', 'Music', 'stat', and 'stat.c'.

The second screenshot shows the source code of 'list.c' in a nano editor. The code is as follows:

```
1  #include <stdio.h>
2  #include <dirent.h>
3
4  int main() {
5      struct dirent
6      **namelist; int n,i;
7      char pathname[100];
8      getcwd(pathname);
9
10     n = scandir(pathname, &namelist, 0, alphasort);
11     if(n < 0)
12         printf("Error\n");
13     else
14         for(i=0; i<n; i++) if(namelist[i]->d_name[0] != '.')
15             printf("%-20s", namelist[i]->d_name);
16 }
```

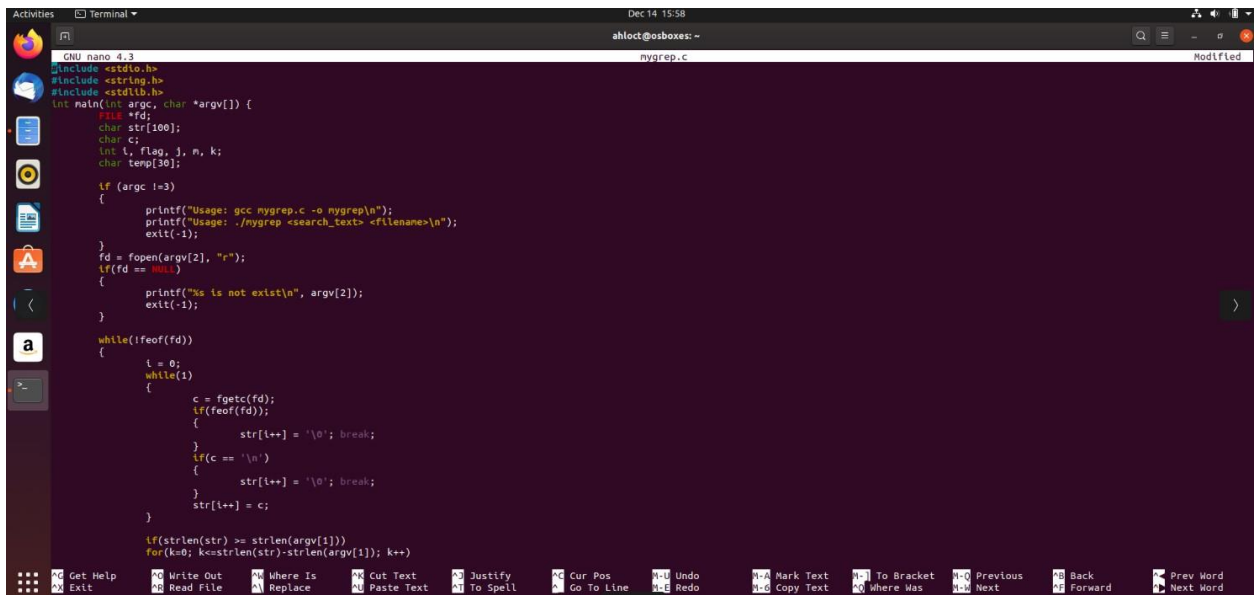
The terminal window title is 'ahloct@osboxes:~' and the file name is 'list.c'.

## 2. Mygrep



A terminal window with a dark purple background. The prompt is `ahloct@osboxes: ~`. The user enters the following commands:

```
ahloct@osboxes:~$ nano mygrep.c
ahloct@osboxes:~$ gcc mygrep.c -o mygrep
ahloct@osboxes:~$ ./mygrep
Usage: gcc mygrep.c -o mygrep
Usage: ./mygrep <search_text> <filename>
ahloct@osboxes:~$ ./mygrep break mygrep.c
ahloct@osboxes:~$
```



A terminal window showing the source code of `mygrep.c` in the `nano` editor. The file is titled `mygrep.c` and is marked as `Modified`. The code is as follows:

```
GNU nano 4.3 mygrep.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

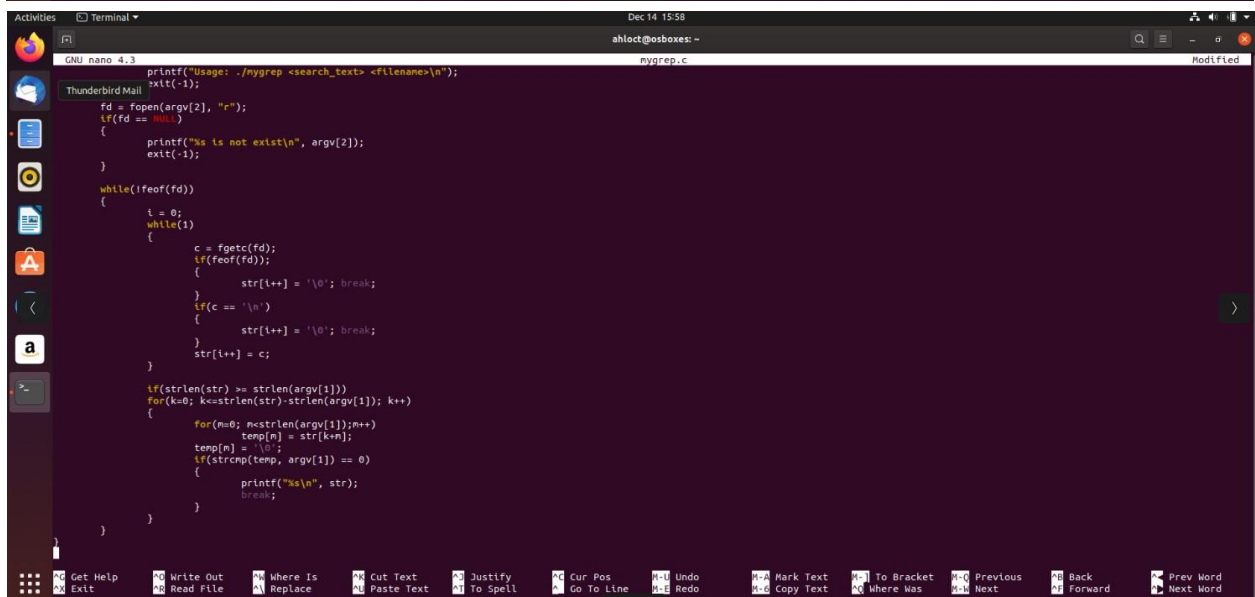
int main(int argc, char *argv[]) {
    FILE *fd;
    char str[100];
    char c;
    int i, flag, j, n, k;
    char temp[30];

    if (argc != 3)
    {
        printf("Usage: gcc mygrep.c -o mygrep\n");
        printf("Usage: ./mygrep <search_text> <filename>\n");
        exit(-1);
    }

    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;
            }
            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++)
            {
                printf("Usage: ./mygrep <search_text> <filename>\n");
                exit(-1);
            }
    }
}
```



A terminal window showing the source code of `mygrep.c` in the `nano` editor. The file is titled `mygrep.c` and is marked as `Modified`. The code is as follows:

```
GNU nano 4.3 mygrep.c
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int main(int argc, char *argv[]) {
    FILE *fd;
    char str[100];
    char c;
    int i, flag, j, n, k;
    char temp[30];

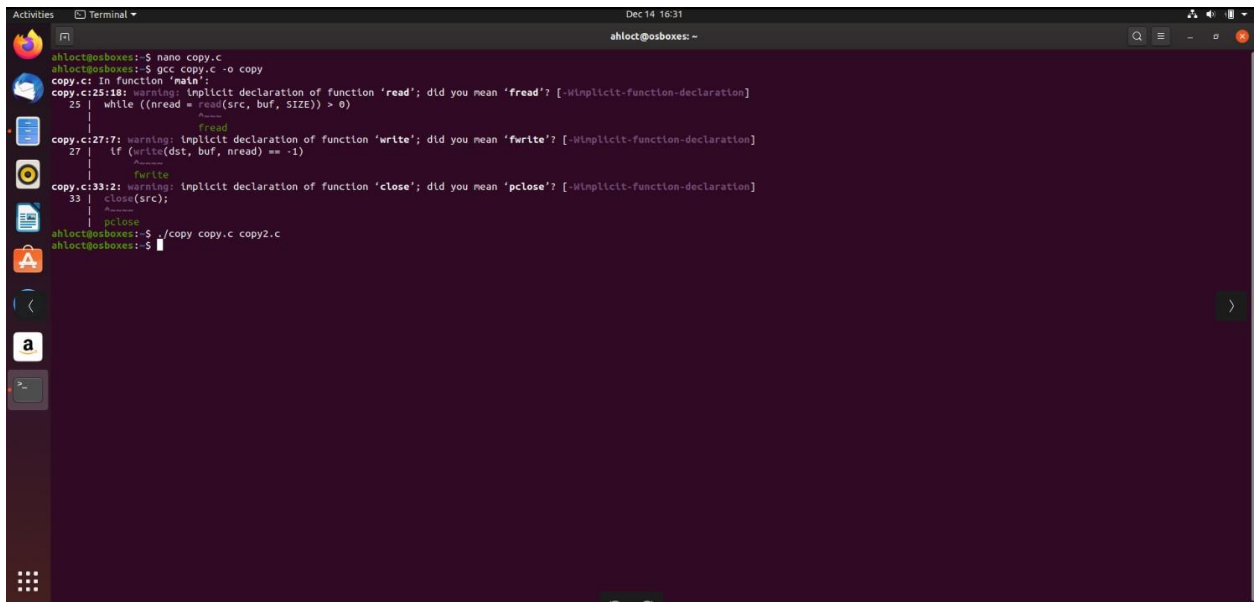
    if (argc != 3)
    {
        printf("Usage: gcc mygrep.c -o mygrep\n");
        printf("Usage: ./mygrep <search_text> <filename>\n");
        exit(-1);
    }

    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;
            }
            if (c == '\n')
            {
                str[i++] = '\0'; break;
            }
            str[i++] = c;
        }

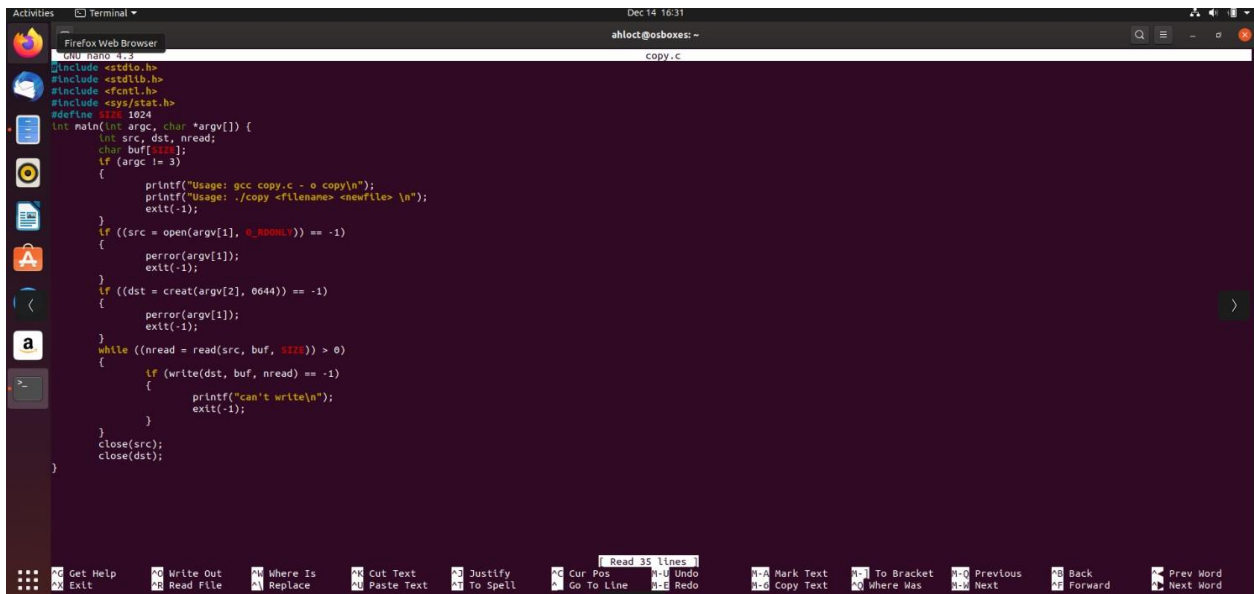
        if (strlen(str) >= strlen(argv[1]))
            for (k=0; k<=strlen(str)-strlen(argv[1]); k++)
            {
                for (n=0; n<=strlen(argv[1]); n++)
                {
                    temp[n] = str[k+n];
                    if (strcmp(temp, argv[1]) == 0)
                    {
        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

### 3. Copy



A terminal window titled 'ahloct@osboxes: -' showing the compilation and execution of a C program. The user runs 'nano copy.c' to edit the file. The code in 'copy.c' includes `<stdio.h>`, `<stdlib.h>`, `<fcntl.h>`, and `<sys/stat.h>`. It defines `SIZE` as 1024 and has a `main` function that takes three arguments: `src`, `dst`, and `nread`. It checks if `argc` is 3, then opens `src` and creates `dst`. It then enters a loop reading from `src` and writing to `dst` until `nread` is 0. Finally, it closes both files. The user compiles the program with `gcc copy.c -o copy` and runs it with `./copy copy.c copy2.c`. The terminal shows warnings about implicit declarations of `read`, `write`, and `close`.

```
ahloct@osboxes:~$ nano 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:33:2: 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:~$
```

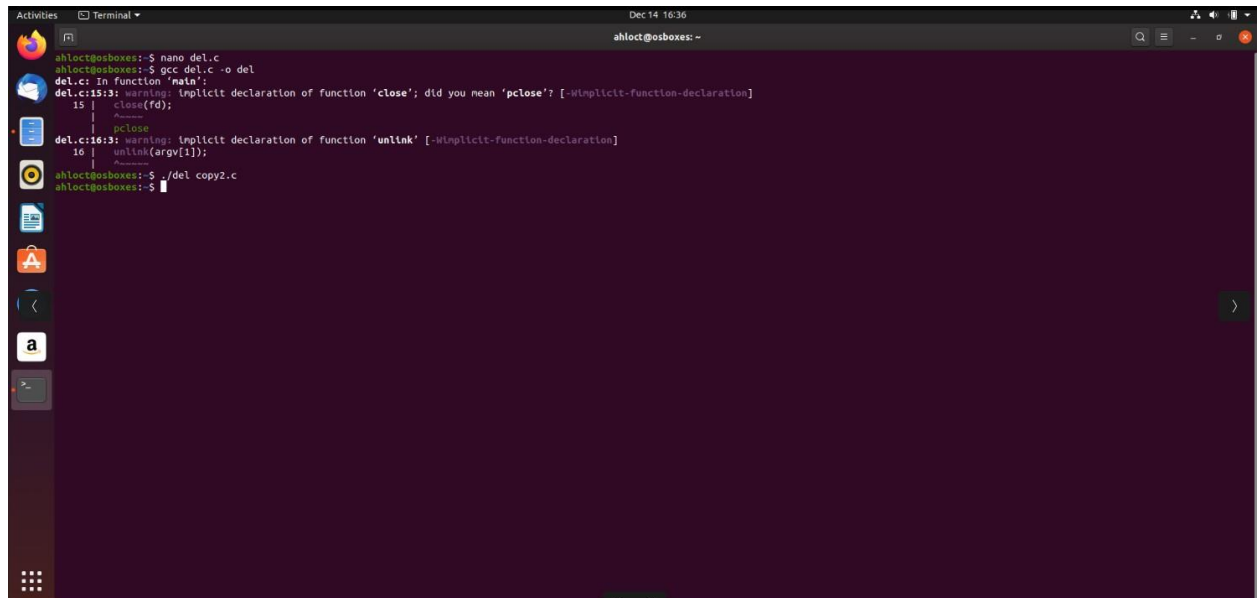


A terminal window titled 'ahloct@osboxes: -' showing the source code of 'copy.c'. The code includes `<stdio.h>`, `<stdlib.h>`, `<fcntl.h>`, and `<sys/stat.h>`. It defines `SIZE` as 1024 and has a `main` function that takes three arguments: `src`, `dst`, and `nread`. It checks if `argc` is 3, then opens `src` and creates `dst`. It then enters a loop reading from `src` and writing to `dst` until `nread` is 0. Finally, it closes both files.

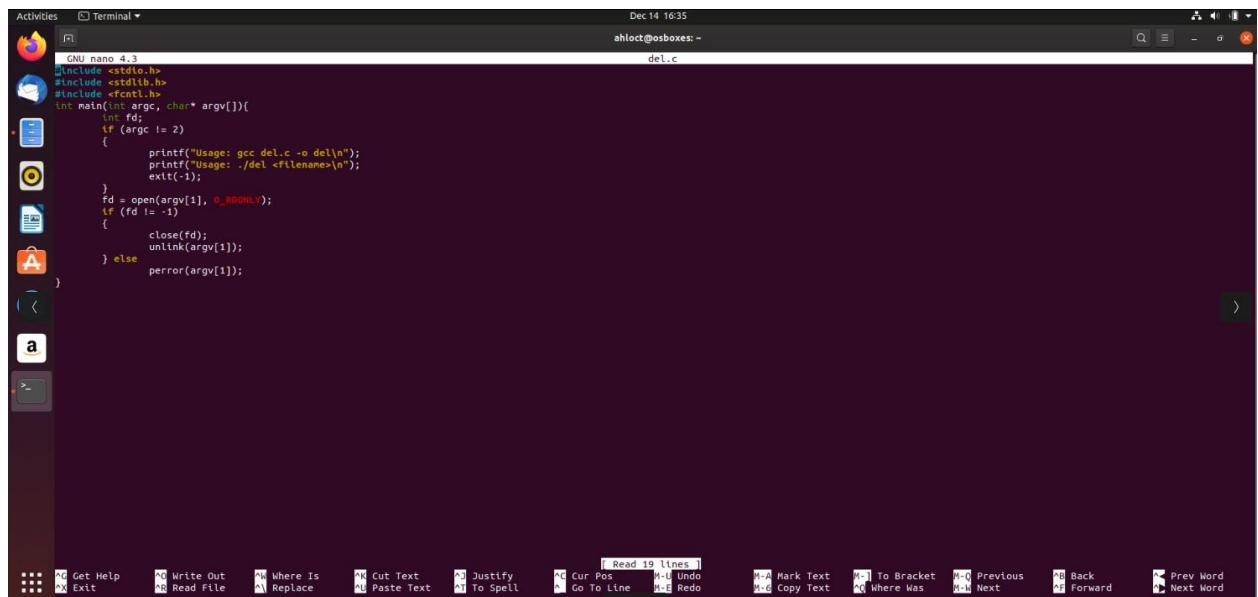
```
copy.c
#include <stdio.h>
#include <stdlib.h>
#include <fcntl.h>
#include <sys/stat.h>
#define SIZE 1024

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[1]);
        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);
}
```

## 4. Del



```
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);
        |     ^~~~~~
        |     pclose
del.c:16:13: warning: implicit declaration of function 'unlink' [-Wimplicit-function-declaration]
    16 |     unlink(argv[1]);
        |     ^~~~~~
ahloct@osboxes:~$ ./del copy2.c
ahloct@osboxes:~$
```



```
GNU nano 4.3
#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]);
}
```