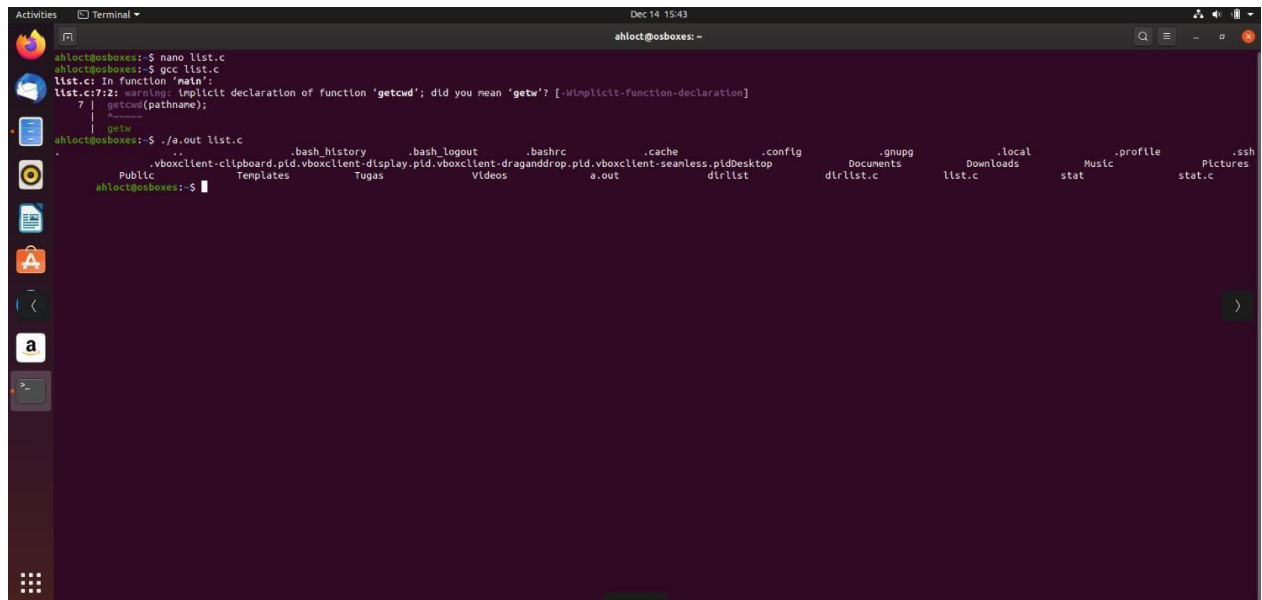


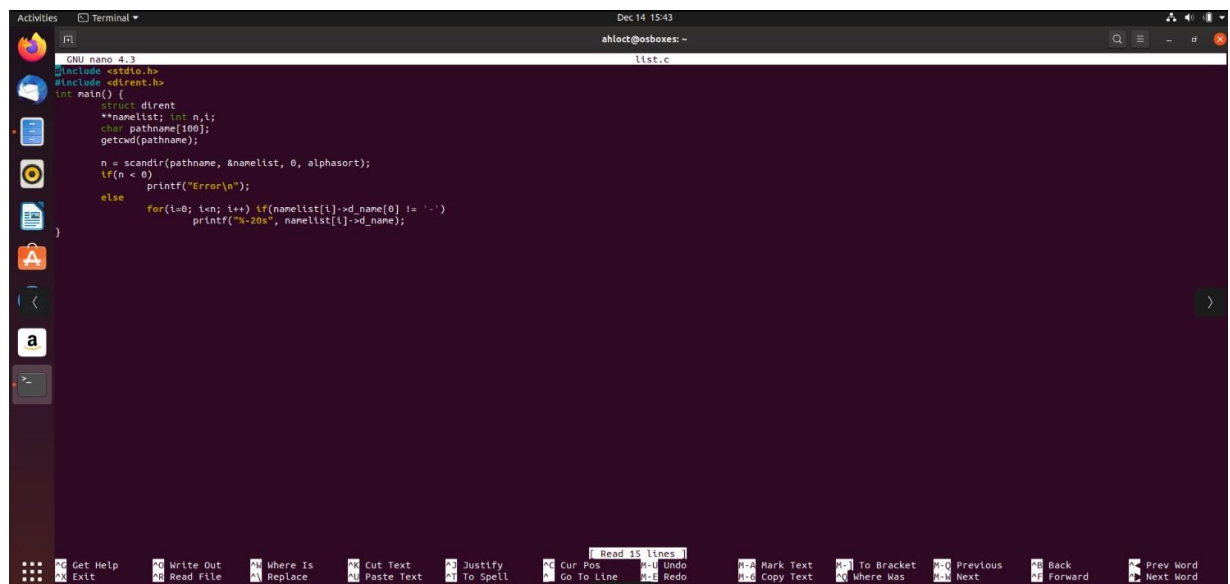
Nama : Farid Aziz priyo F
NIM : L200180117
Kelas : C

Modul 10

1. List



```
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
Public
Templates
Videos
a.out
dirlist
dirlist.c
list.c
stat
stat.c
```



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

2. Mygrep

```
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:~$
```

```
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("File 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++)
            {
                if (strcmp(str+k, argv[1]) == 0)
                {
                    printf("%s\n", str);
                    break;
                }
            }
        }
    }
}
```

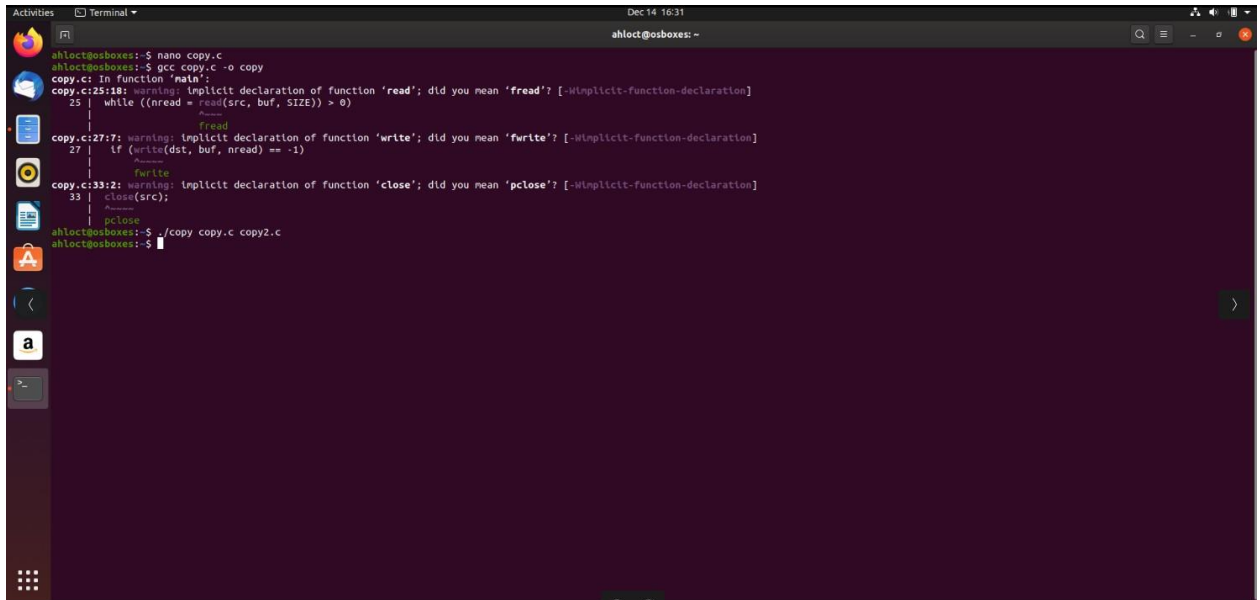
```
GNU nano 4.3 mygrep.c
printf("Usage: ./mygrep <search_text> <filename>\n");
exit(-1);

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

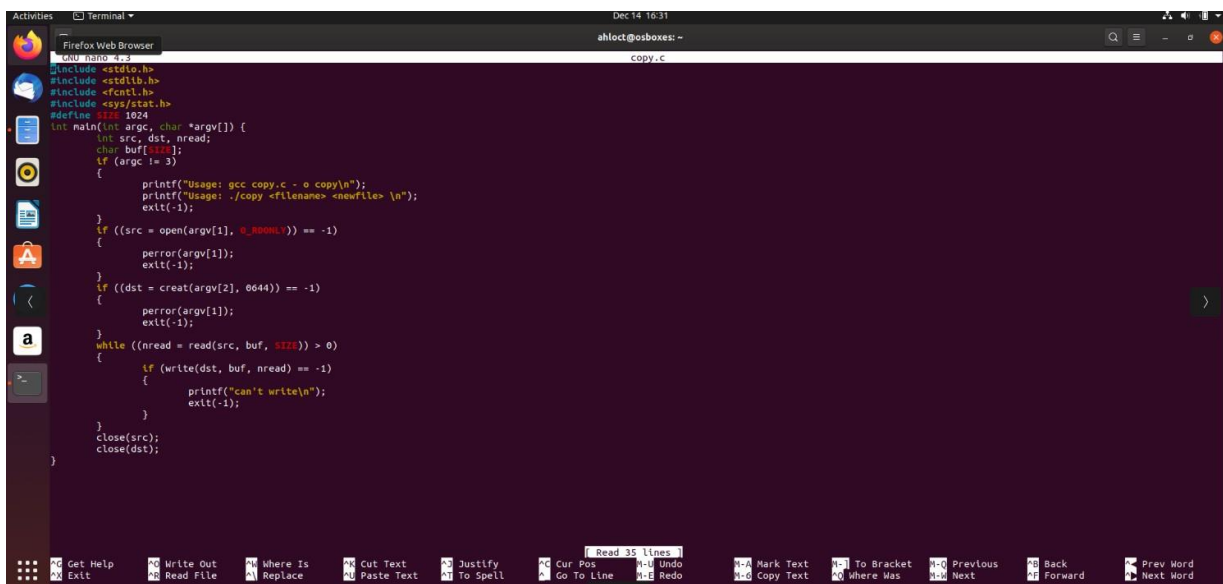
while (!feof(fd))
{
    i = 0;
    while (1)
    {
        c = fgetc(fd);
        if (feof(fd))
        {
            str[i++] = '\0'; break;
        }
        if (c == '\n')
        {
            str[i++] = '\0'; break;
        }
        str[i++] = c;
    }

    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



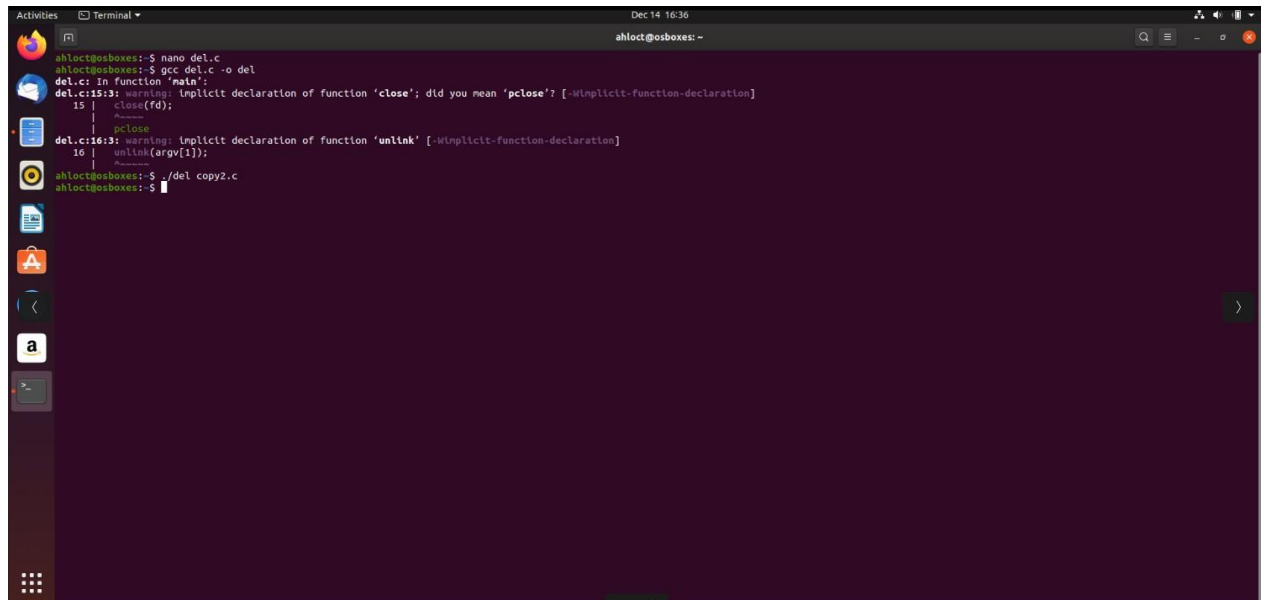
```
ahloct@osboxes:~$ nano copy.c
ahloct@osboxes:~$ gcc copy.c -o copy
copy.c: In function 'main':
copy.c:25:18: warning: implicit declaration of function 'read'; did you mean 'fread'? [-Wimplicit-function-declaration]
   25 |     while ((nread = read(src, buf, SIZE)) > 0)
      |                  ^~~~~
      |                  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:~$
```



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

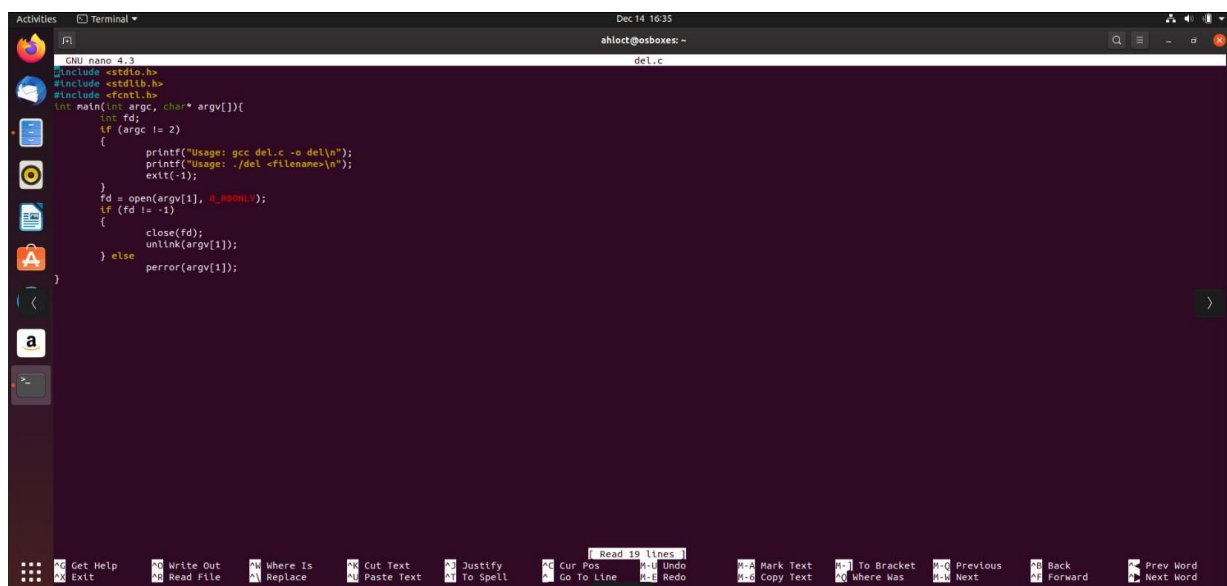
int main(int argc, char *argv[]) {
    int src, dst, nread;
    char buf[SIZE];
    if (argc != 3)
    {
        printf("Usage: gcc copy.c -o copy\n");
        printf("Usage: ./copy <filename> <newfile> \n");
        exit(-1);
    }
    if ((src = open(argv[1], O_RDONLY)) == -1)
    {
        perror(argv[1]);
        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



A terminal window titled 'Terminal' with the date 'Dec 14 16:36' and the user 'ahloct@osboxes:'. The user enters the following commands:

```
ahloct@osboxes:~$ nano del.c
ahloct@osboxes:~$ gcc del.c -o del
del.c: In function 'main':
del.c:15:3: warning: implicit declaration of function 'close'; did you mean 'pclose'? [-Wimplicit-function-declaration]
    15 |     close(fd);
        |     ^~~~~~
        |     pclose
del.c:16:3: warning: implicit declaration of function 'unlink' [-Wimplicit-function-declaration]
    16 |     unlink(argv[1]);
        |
ahloct@osboxes:~$ ./del copy2.c
ahloct@osboxes:~$
```



A terminal window titled 'Terminal' with the date 'Dec 14 16:35' and the user 'ahloct@osboxes:'. The user enters the command 'cat del.c', and the contents of the file are displayed:

```
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]);
}
```

At the bottom of the terminal, a menu bar is visible with the following options: Get Help, Exit, Write Out, Read File, Where Is, Replace, Cut Text, Paste Text, Justify, To Spell, Cur Pos, Go To Line, Undo, Redo, Mark Text, Copy Text, To Bracket, Where Was, Previous, Next, Back, Forward, Prev Word, Next Word.