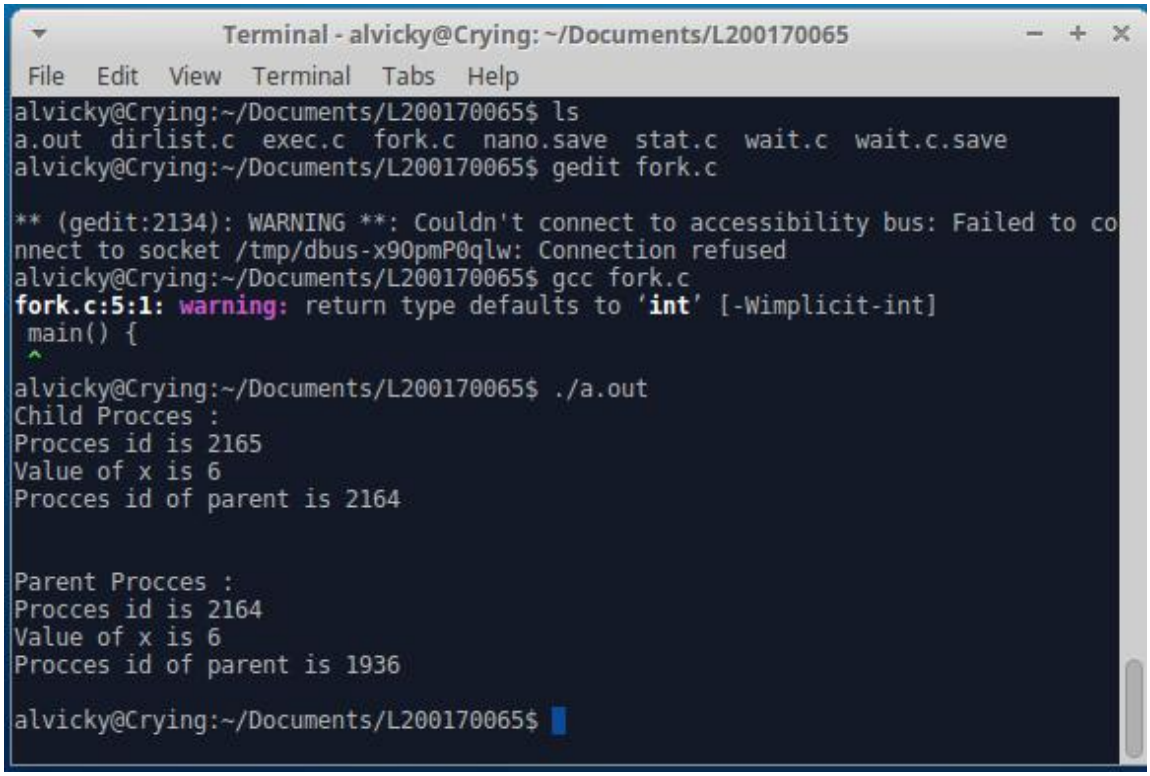
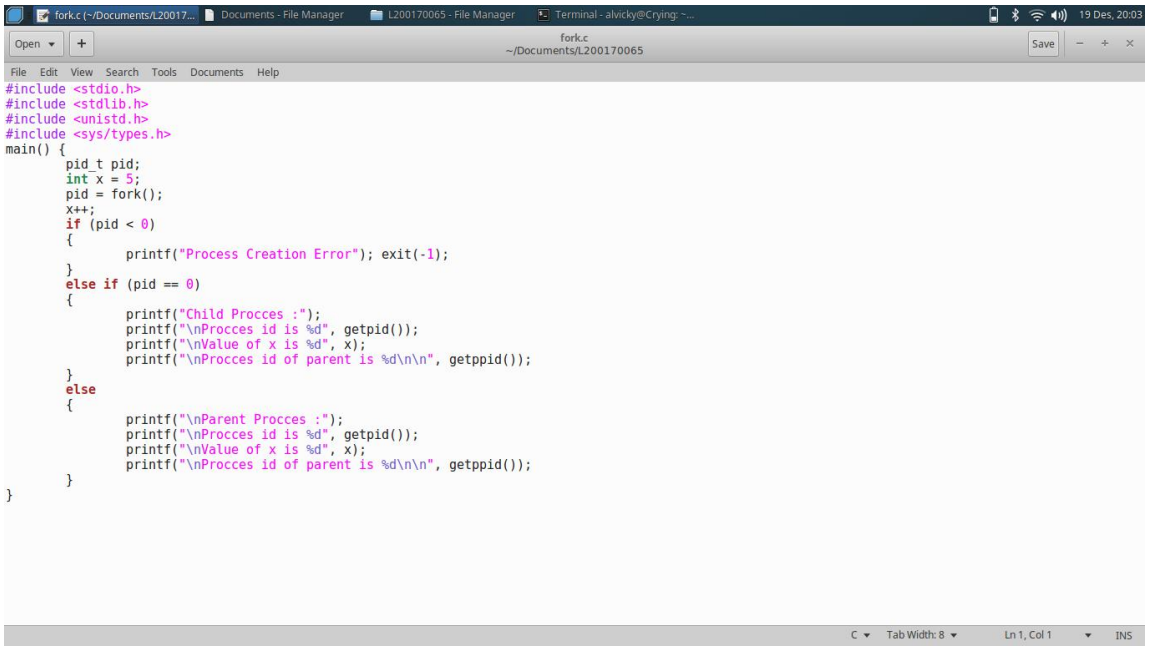
https://github.com/L200170065

Nama : Muhammad Vicky Al Hasri
NIM : L200170065
Kelas : C
Matkul : Praktikum Sistem Operasi

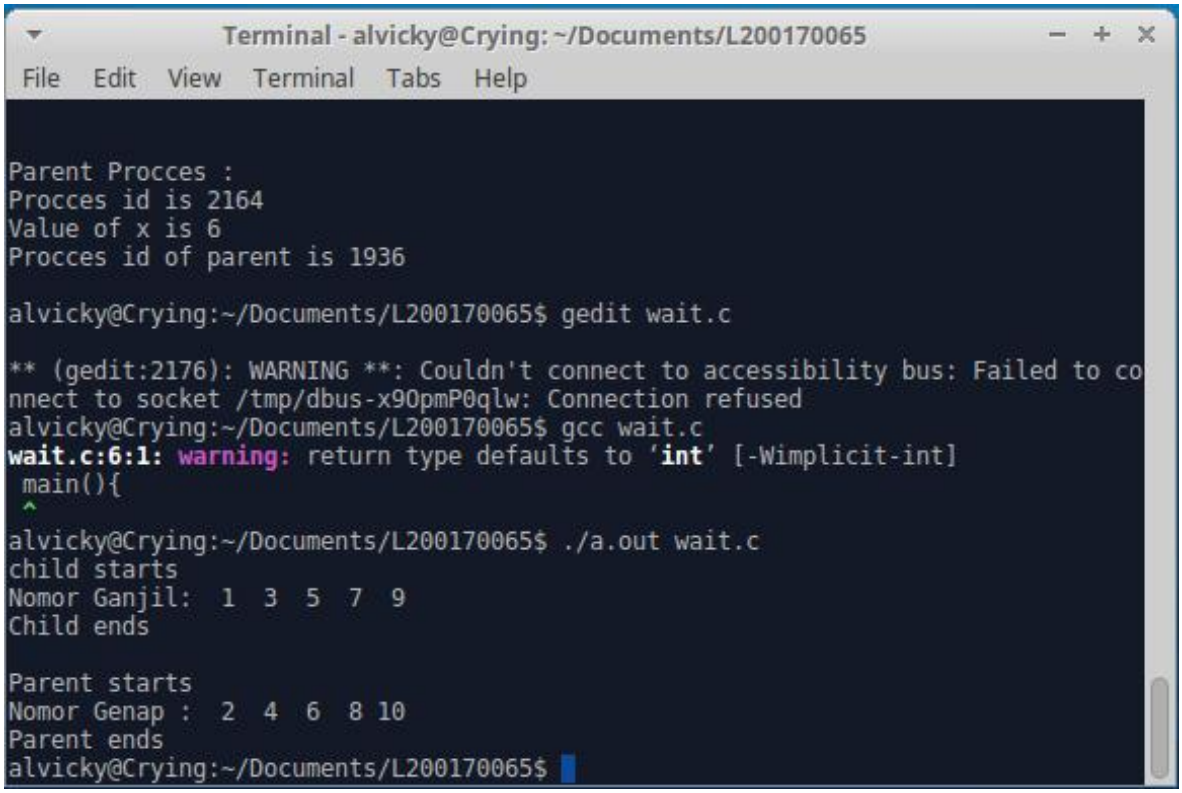
Fork.c



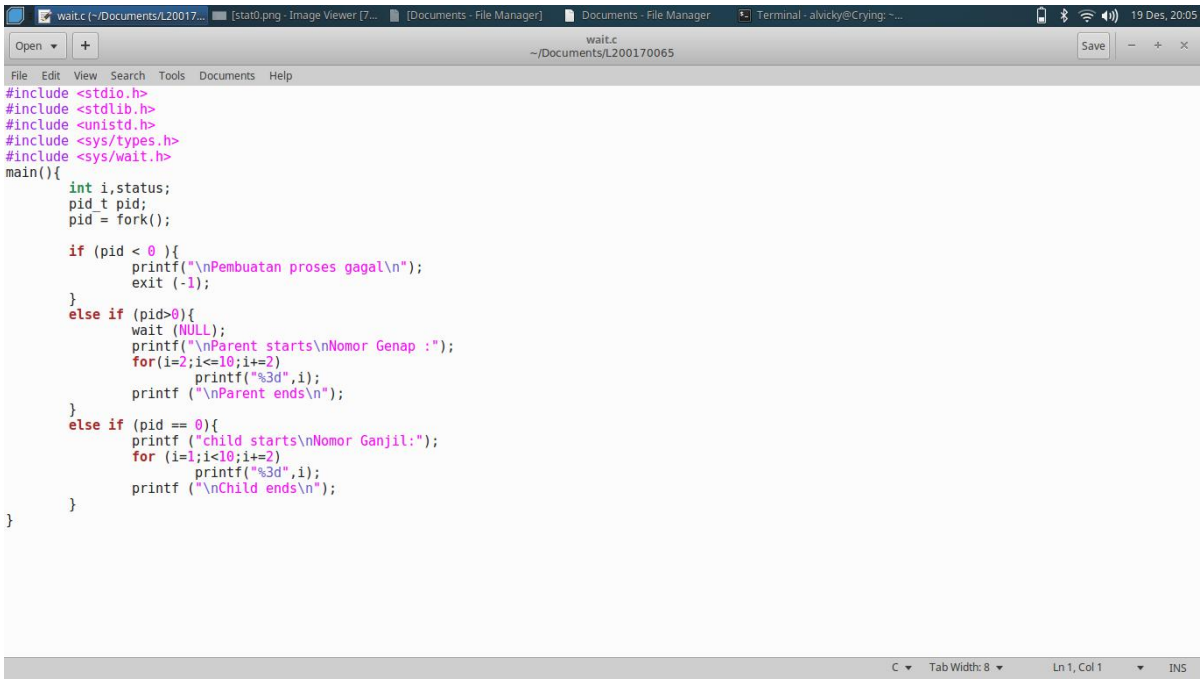
Source Code Fork.c



Wait.c

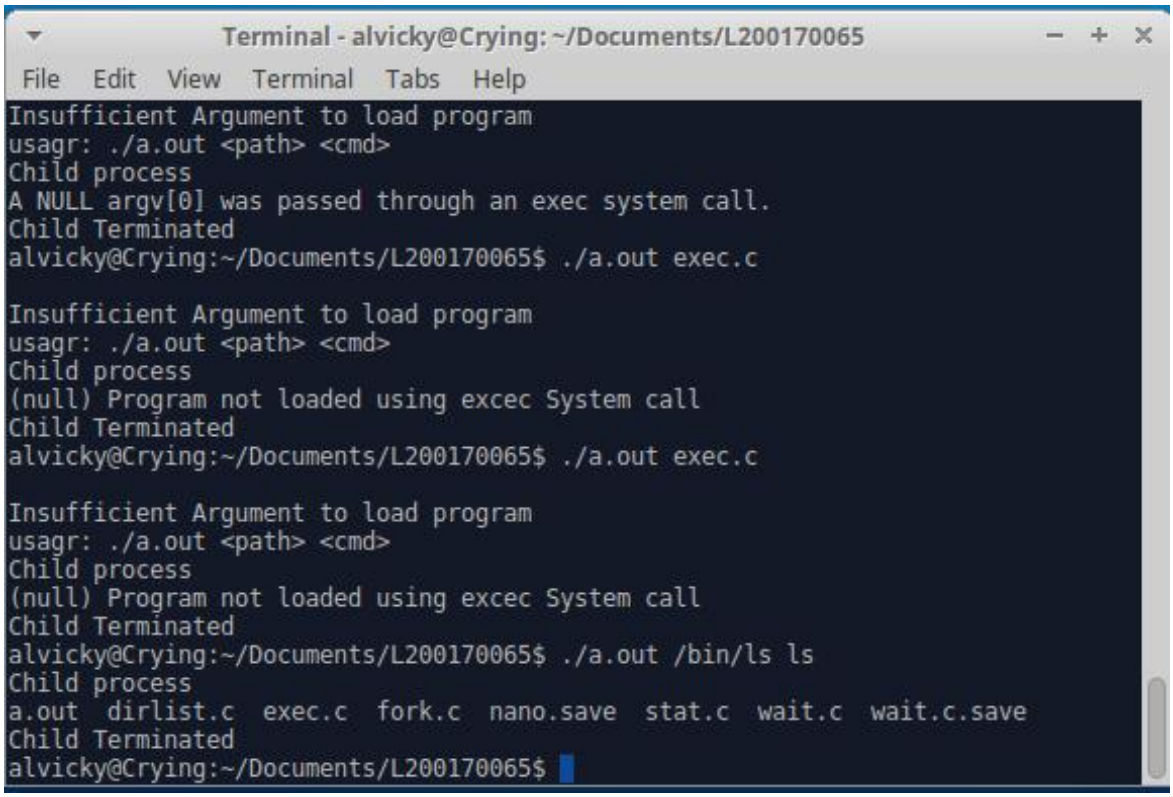


Source Code Wait.c

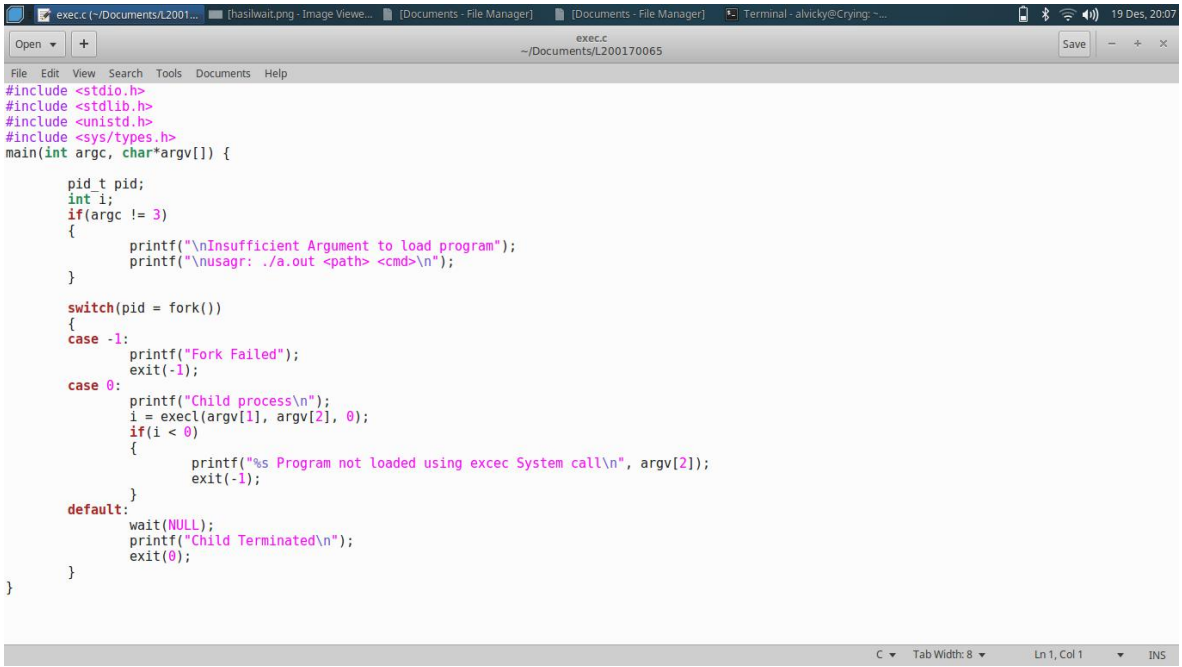


https://github.com/L200170065

Exec.c



Source Code Exec.c



https://github.com/L200170065

Stat.c

```
Terminal - alvicky@Crying: ~/Documents/L200170065
File  Edit  View  Terminal  Tabs  Help
 has type '__ino_t {aka long unsigned int}' [-Wformat=]
  printf("Inode no. : %d\n", file.st_ino);
          ^
stat.c:24:9: 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);
          ^
stat.c:25:9: 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);
          ^
alvicky@Crying:~/Documents/L200170065$ ./a.out stat.c
User id :1000
Groupid :1000
Block size : 4096
Blocks allocated : 8
Inode no. : 534149
last accessed : Wed Dec 19 20:11:12 2018
last modified : Fri Nov 30 10:35:36 2018
File size : 1372 bytes
No. of links : 1
Permissions : -rw-r--r--
File type : regular
alvicky@Crying:~/Documents/L200170065$
```

Source Code Stat.c

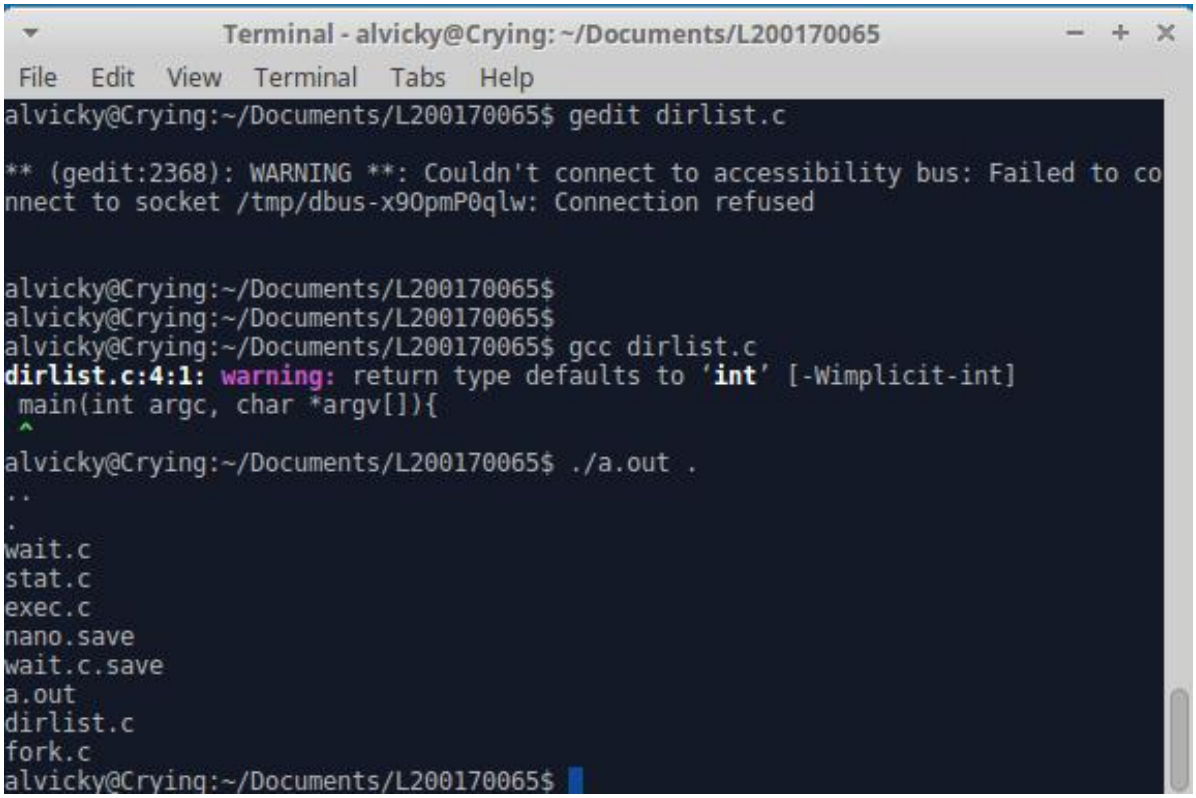
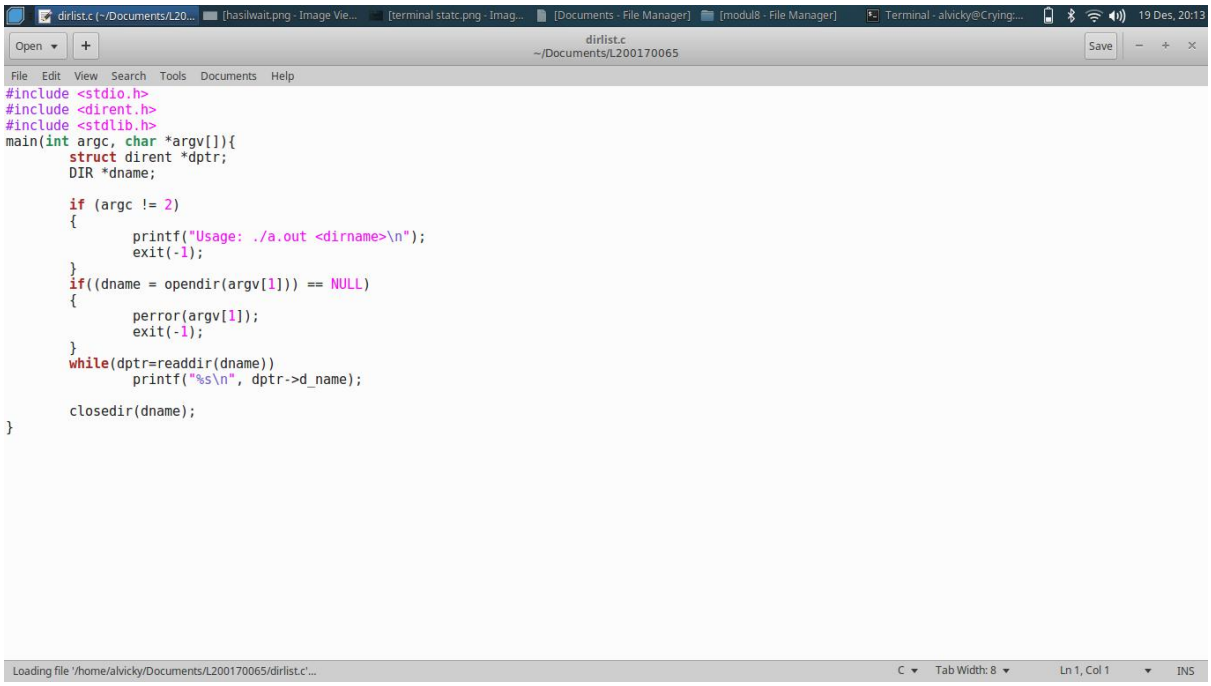
```
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <time.h>
int main(int argc, char*argv[]) {
        struct stat
        file; int n;
        if (argc != 2)
        {
                printf("Usage :./a.out <filename>\n"); exit(-1);
        }
        if ((n = stat(argv[1], &file)) == -1)
        {
                perror(argv[1]);
                exit(-1);
        }
        printf("User id :%d\n", file.st_uid);
        printf("Groupid :%d\n", file.st_gid);
        printf("Block size : %d\n", file.st_blksize);
        printf("Blocks 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("Permissions : ");
        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("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("Groupid :%d\n", file.st_gid);
        printf("Block size : %d\n", file.st_blksize);
        printf("Blocks 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("Permissions : ");
        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");


}
```

https://github.com/L200170065

Dirlist.c



Source Code Dirlist.c