**Nama** : Resky Budi Nugroho
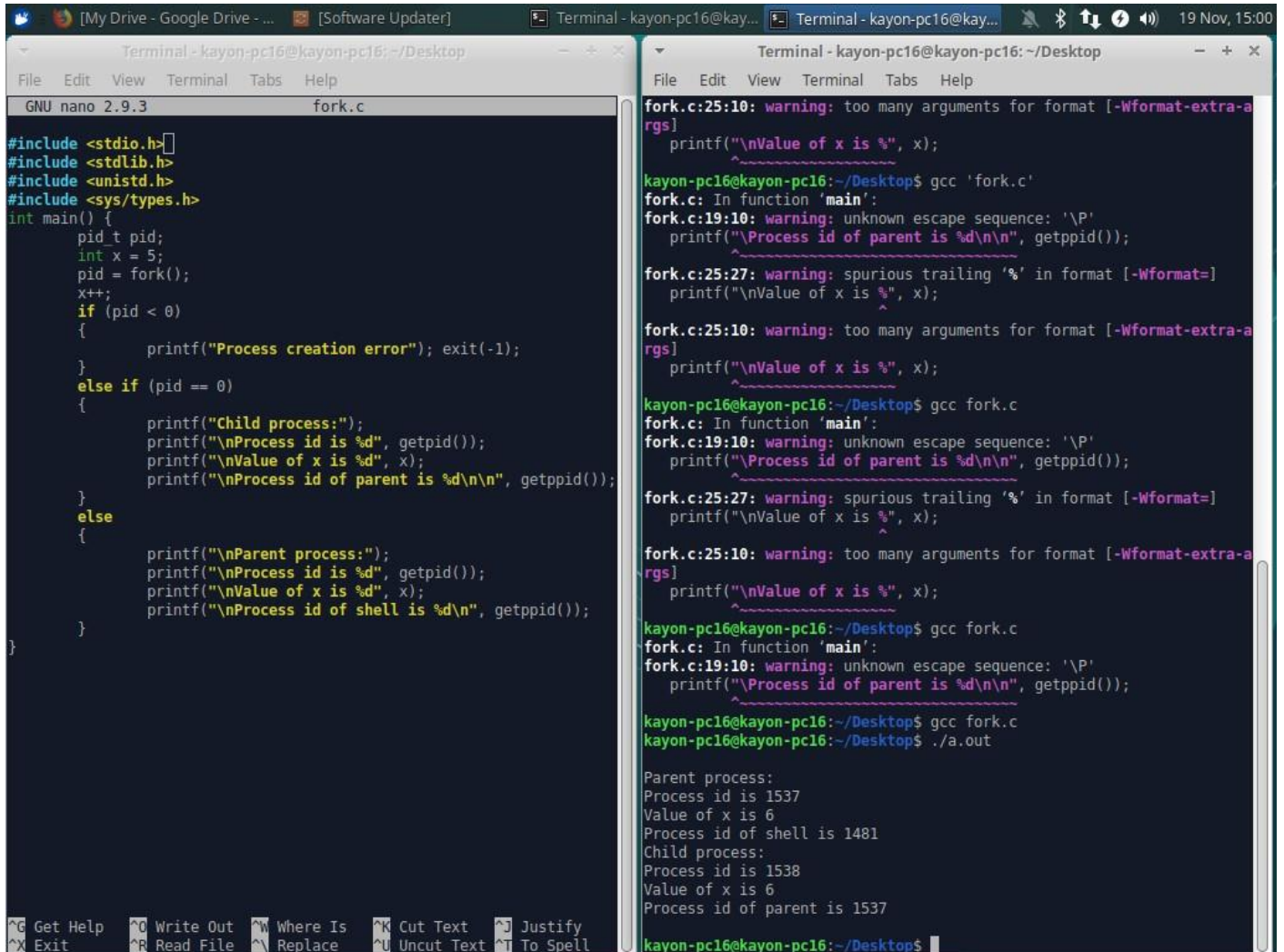
**NIM** : L200180018

**Kelas** : A

# MODUL 8

## 1. Membuat child "Child Process" dengan menggunakan System Call "Fork"
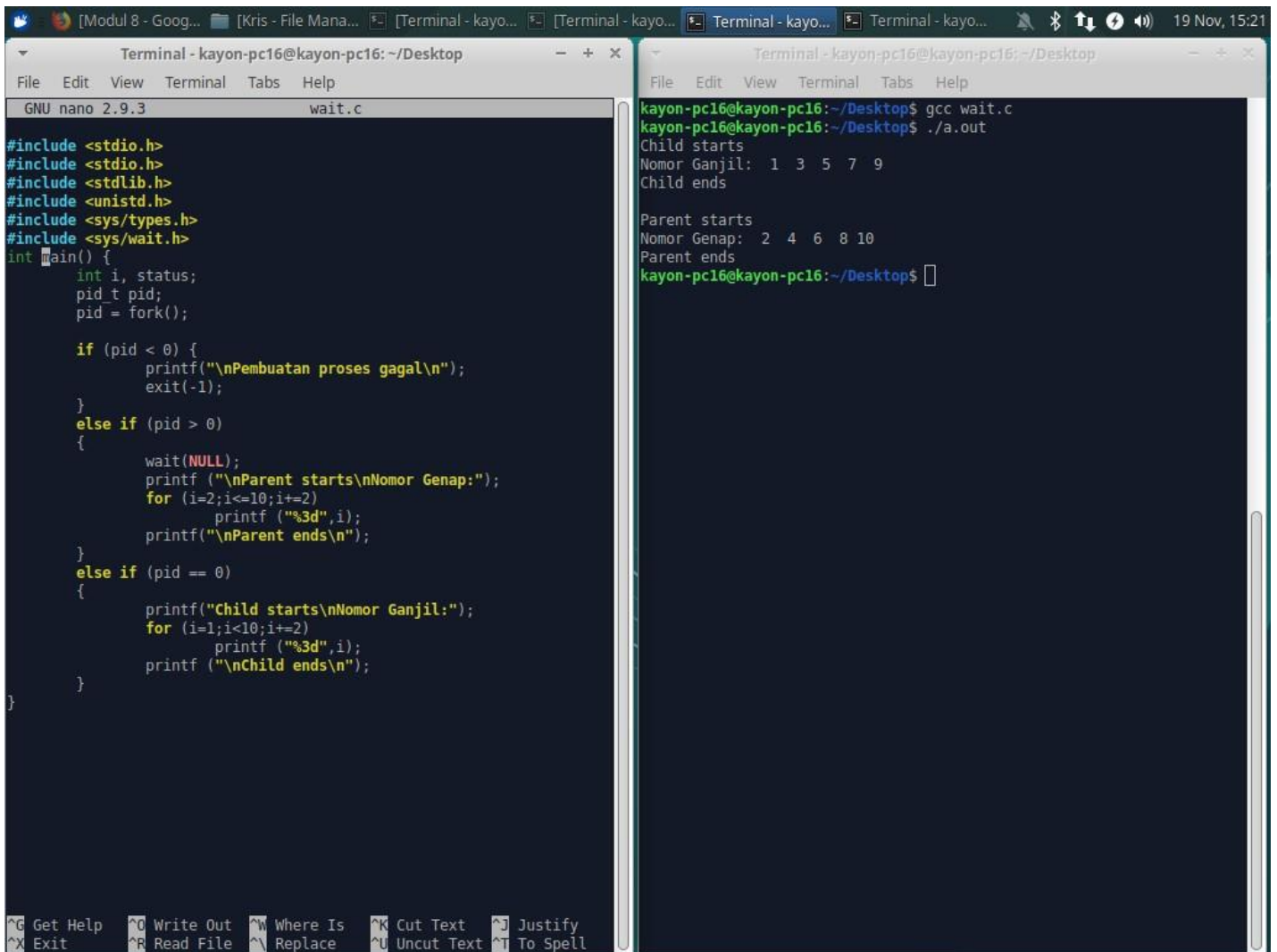


## 2. Menghentikan sementara (block) proses parent dengan proses child selesai, menggunakan system call "wait"

GNU nano 2.9.3                    wait.c

```c
#include <stdio.h>
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
int main() {
        int i, status;
        pid_t pid;
        pid = fork();

        if (pid < 0) {
                printf("\nPembuatan proses gagal\n");
                exit(-1);
        }
        else if (pid > 0)
        {
                wait(NULL);
                printf ("\nParent starts\nNomor Genap:");
                for (i=2;i<=10;i+=2)
                        printf ("%3d",i);
                printf("\nParent ends\n");
        }
        else if (pid == 0)
        {
                printf("Child starts\nNomor Ganjil:");
                for (i=1;i<10;i+=2)
                        printf ("%3d",i);
                printf ("\nChild ends\n");
        }
}
```

Terminal output:

```
kayon-pc16@kayon-pc16:~/Desktop$ gcc wait.c
kayon-pc16@kayon-pc16:~/Desktop$ ./a.out
Child starts
Nomor Ganjil:  1  3  5  7  9
Child ends

Parent starts
Nomor Genap:  2  4  6  8 10
Parent ends
kayon-pc16@kayon-pc16:~/Desktop$ 
```

**3. Loading Program yang dapat dieksekusi dalam sebuah "Child" Proses menggunakan perintah system call "exec"**

File   Edit   View   Terminal   Tabs   Help

root@kayon-pc25: /home/kay...  ✕ | kayon-pc25@kayon-pc25: ~  ✕ | root@kayon-pc25: /home/kay...  ✕ | kayon-pc25@kayon-pc25: ~  ✕ | root@kayon-pc25: /home/kay...  ✕ | kayon-pc25@kayon-pc25: ~  ✕

```
kayon-pc25@kayon-pc25:~$ gcc exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
 main(int argc, char*argv[]) {
 ^~~~
exec.c: In function 'main':
exec.c:23:3: warning: missing sentinel in function call [-Wformat=]
   i = execl(argv[1], argv[2], 0);
   ^
exec.c:30:3: warning: implicit declaration of function 'wait'; did you mean 'main'? [-Wimplicit-function-declaration]
   wait(NULL);
   ^~~~
   main
kayon-pc25@kayon-pc25:~$ ./a.out /bin/ls ls
Child process
a.out  Desktop  Documents  Downloads  exec.c  fork.c  infol200180110.sh.save  infol200180110.sh.save.1  Music  Pictures  Public  Templates  Videos  wait.c
Child Terminated
kayon-pc25@kayon-pc25:~$ gcc exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
 main(int argc, char*argv[]) {
 ^~~~
exec.c: In function 'main':
exec.c:20:3: warning: missing sentinel in function call [-Wformat=]
   i = execl(argv[1], argv[2], 0);
   ^
exec.c:27:3: warning: implicit declaration of function 'wait'; did you mean 'main'? [-Wimplicit-function-declaration]
   wait(NULL);
   ^~~~
   main
kayon-pc25@kayon-pc25:~$ gcc exec.c
exec.c:5:10: fatal error: sys/wait.c: No such file or directory
 #include <sys/wait.c>
          ^~~~~~~~~~~~
compilation terminated.
kayon-pc25@kayon-pc25:~$ gcc exec.c
exec.c:6:1: warning: return type defaults to 'int' [-Wimplicit-int]
 main(int argc, char*argv[]) {
 ^~~~
kayon-pc25@kayon-pc25:~$ gcc exec.c
kayon-pc25@kayon-pc25:~$ ./a.out /bin/ls ls
Child process
a.out  Desktop  Documents  Downloads  exec.c  fork.c  infol200180110.sh.save  infol200180110.sh.save.1  Music  Pictures  Public  Templates  Videos  wait.c
Child Terminated
kayon-pc25@kayon-pc25:~$
```
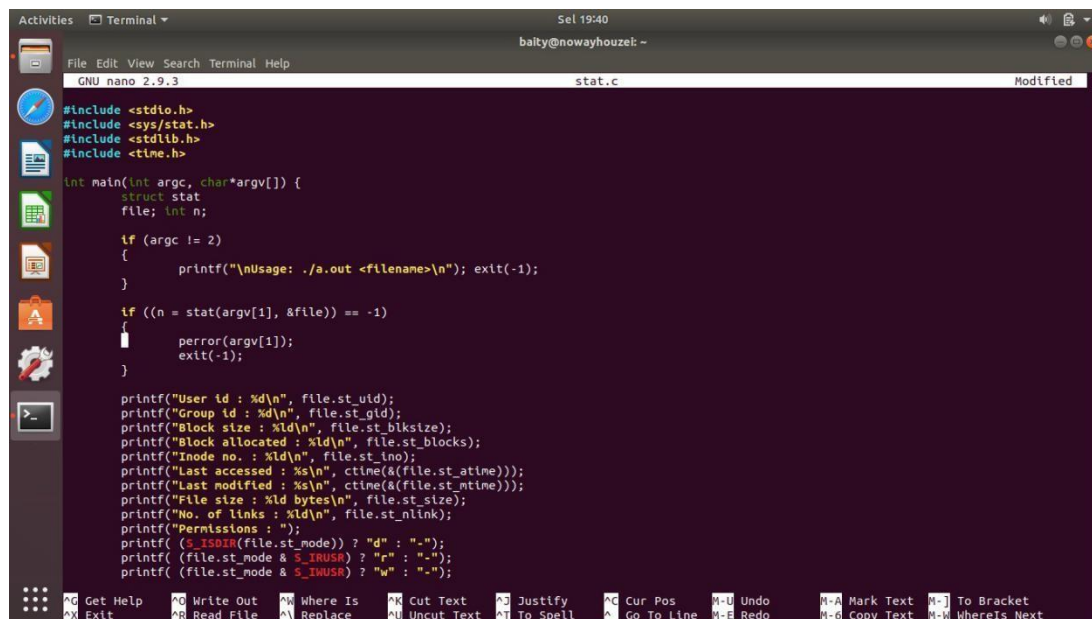
Untitled 1 - LibreOffice Writer   [Software Updater]   Terminal - kayon-pc25@kayo...        19 Nov, 15:50

---

File   Edit   View   Terminal   Tabs   Help

root@kayon-pc25: /home/kay...  ✕ | kayon-pc25@kayon-pc25: ~  ✕ | root@kayon-pc25: /home/kay...  ✕ | kayon-pc25@kayon-pc25: ~  ✕ | root@kayon-pc25: /home/kay...  ✕ | kayon-pc25@kayon-pc25: ~  ✕

```
  GNU nano 2.9.3                                    exec.c

#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <sys/wait.h>
int main(int argc, char*argv[]) {
        pid_t pid;
        int i;
        if (argc != 3)
        {
                printf("\nInsufficient arguments to load program");
                printf("\nUsage: ./a.out <path> <cmd>\n"); exit(-1);
        }
        switch(pid = fork())
        {
        case -1:
                printf("Fork failed");
                exit(-1);
        case 0:
                printf("Child process\n");
                i = execl(argv[1], argv[2], NULL);
                if (i < 0)
                {
                        printf("%s program not loaded using exec system call\n", argv[2]);
                        exit(-1);
                }
        default:
                wait(NULL);
                printf("Child Terminated\n");
                exit(-1);
        }
}
```

                                          [ Wrote 32 lines ]
^G Get Help    ^O Write Out    ^W Where Is    ^K Cut Text     ^J Justify     ^C Cur Pos     M-U Undo    M-A Mark Text   M-] To Bracket   M-▲ Previous    ^B Back
^X Exit        ^R Read File    ^\ Replace     ^U Uncut Text   ^T To Spell    ^_ Go To Line  M-E Redo    M-6 Copy Text   M-W WhereIs Next M-▼ Next        ^F Forward

Untitled 1 - LibreOffice Writer   [Software Updater]   Terminal - root@kayon-pc25: ...        19 Nov, 15:50

## 4. Menambahkan status file menggunakan perintha system call "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("\nUsage: ./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("Group id : %d\n", file.st_gid);
        printf("Block size : %ld\n", file.st_blksize);
        printf("Block allocated : %ld\n", file.st_blocks);
        printf("Inode no. : %ld\n", file.st_ino);
        printf("Last accessed : %s\n", ctime(&(file.st_atime)));
        printf("Last modified : %s\n", ctime(&(file.st_mtime)));
        printf("File size : %ld bytes\n", file.st_size);
        printf("No. of links : %ld\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" : "-");
```
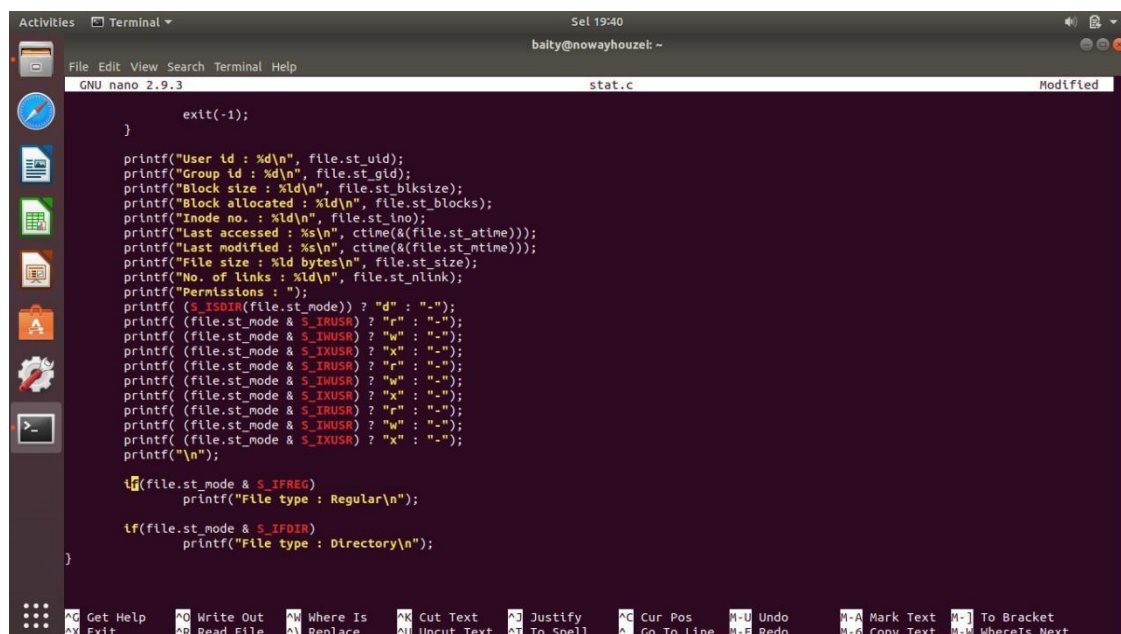
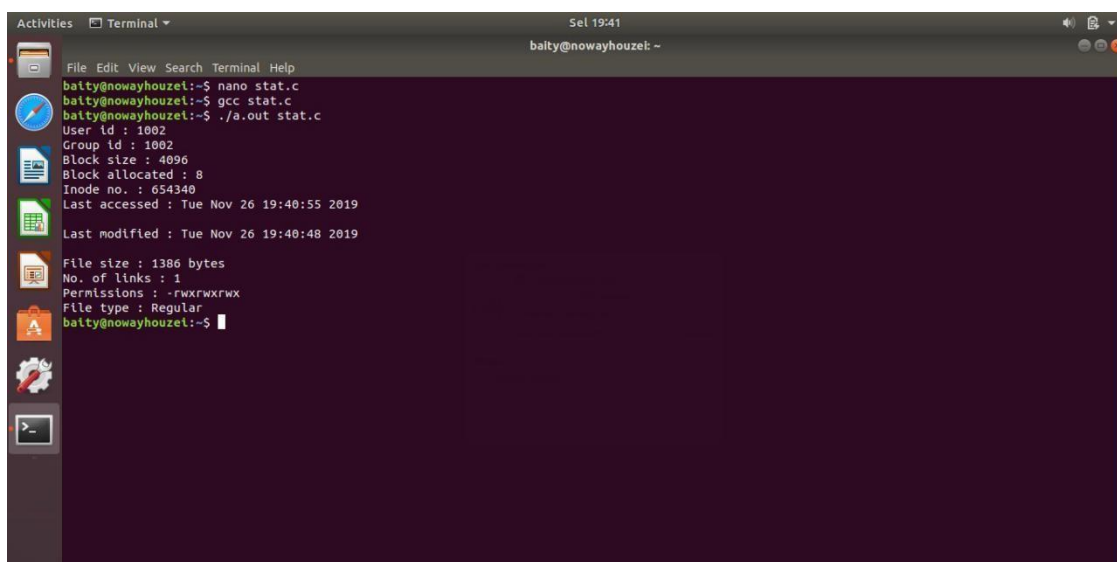

```c
                exit(-1);
        }

        printf("User id : %d\n", file.st_uid);
        printf("Group id : %d\n", file.st_gid);
        printf("Block size : %ld\n", file.st_blksize);
        printf("Block allocated : %ld\n", file.st_blocks);
        printf("Inode no. : %ld\n", file.st_ino);
        printf("Last accessed : %s\n", ctime(&(file.st_atime)));
        printf("Last modified : %s\n", ctime(&(file.st_mtime)));
        printf("File size : %ld bytes\n", file.st_size);
        printf("No. of links : %ld\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_IRUSR) ? "r" : "-");
        printf( (file.st_mode & S_IWUSR) ? "w" : "-");
        printf( (file.st_mode & S_IXUSR) ? "x" : "-");
        printf( (file.st_mode & S_IRUSR) ? "r" : "-");
        printf( (file.st_mode & S_IWUSR) ? "w" : "-");
        printf( (file.st_mode & S_IXUSR) ? "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");
}
```
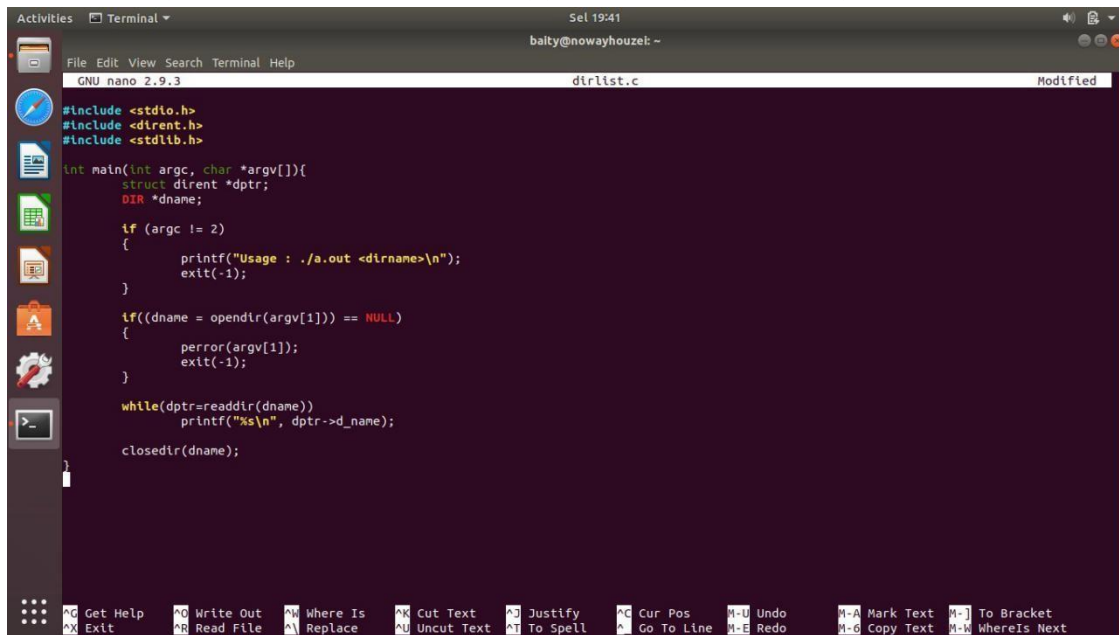


```
baity@nowayhouzei:~$ nano stat.c
baity@nowayhouzei:~$ gcc stat.c
baity@nowayhouzei:~$ ./a.out stat.c
User id : 1002
Group id : 1002
Block size : 4096
Block allocated : 8
Inode no. : 654340
Last accessed : Tue Nov 26 19:40:55 2019

Last modified : Tue Nov 26 19:40:48 2019

File size : 1386 bytes
No. of links : 1
Permissions : -rwxrwxrwx
File type : Regular
baity@nowayhouzei:~$
```

## 5. Menampilkan isi direktori menggunakan perintah system call "readdir"



```c
#include <stdio.h>
#include <dirent.h>
#include <stdlib.h>

int main(int argc, char *argv[]){
        struct dirent *dptr;
        DIR *dname;

        if (argc != 2)
        {
                printf("Usage : ./a.out <dirname>\n");
                exit(-1);
        }

        if((dname = opendir(argv[1])) == NULL)
        {
                perror(argv[1]);
                exit(-1);
        }

        while(dptr=readdir(dname))
                printf("%s\n", dptr->d_name);

        closedir(dname);
}
```
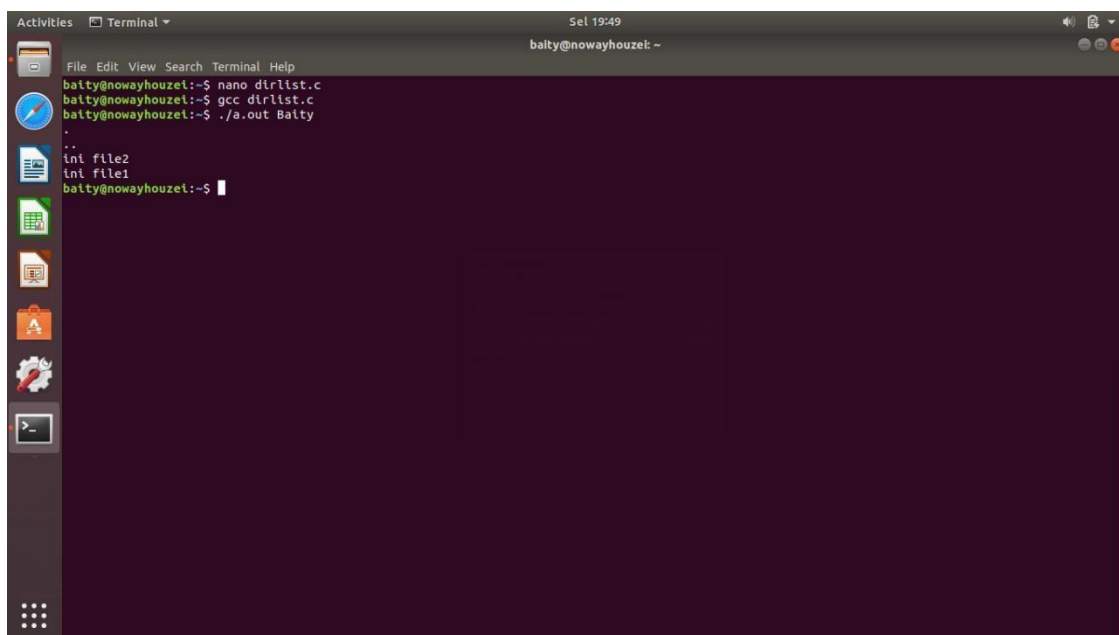


```
baity@nowayhouzei:~$ nano dirlist.c
baity@nowayhouzei:~$ gcc dirlist.c
baity@nowayhouzei:~$ ./a.out Baity
.
..
ini file2
ini file1
baity@nowayhouzei:~$
```