

Rayyan Hanugrah

L200170109

Kelas E

Fork.c

```
My Linux [Berjalan] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
ubuntu@ubuntu:~$ gcc fork.c
gcc: error: fork.c: No such file or directory
gcc: fatal error: no input files
compilation terminated.
ubuntu@ubuntu:~$ nano fork.c
```

```
My Linux [Berjalan] - Oracle VM VirtualBox
GNU nano 2.5.3 File: fork.c
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
main(){
    pid_t pid;
    int x = 5;
    pid = fork();
    x++;
    if (pid < 0)
    {
        printf("Process creation error"); exit(-1);
    }
    else if (pid == 0)
    {
        printf("Child process:");
        printf("\nProcess id is %d", getpid());
        printf("\nValue of x is %d", x);
        printf("\nProcess id of parent is %d\n", getppid());
    }
    else
    {
        printf("\nParent process:");
        printf("\nProcess id is %d", getpid());
        printf("\nValue of x is %d", x);
        printf("\nProcess id of shell is %d\n", getppid());
    }
}
```

```
ubuntu@ubuntu:~$ nano fork.c
ubuntu@ubuntu:~$ gcc fork.c -o fork
ubuntu@ubuntu:~$ ./fork
Parent process:
Process id is 4573
Value of x is 6
Process id of parent is 4460
ubuntu@ubuntu:~$
```

Wait.c

```
My Linux [Berjalan] - Oracle VM VirtualBox
Terminal File Edit View Search Terminal Help
ubuntu@ubuntu:~$ nano wait.c
```

My Linux [Berjalan] - Oracle VM VirtualBox

Terminal File Edit View Search Terminal Help

GNU nano 2.5.3 File: wait.c

```
#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("\nChild starts\nNomor Ganjil:");
        for (i=1;i<=10;i+=2)
            printf(" %3d",i);
        printf("\nChild ends\n");
    }
}
```

Read 29 lines

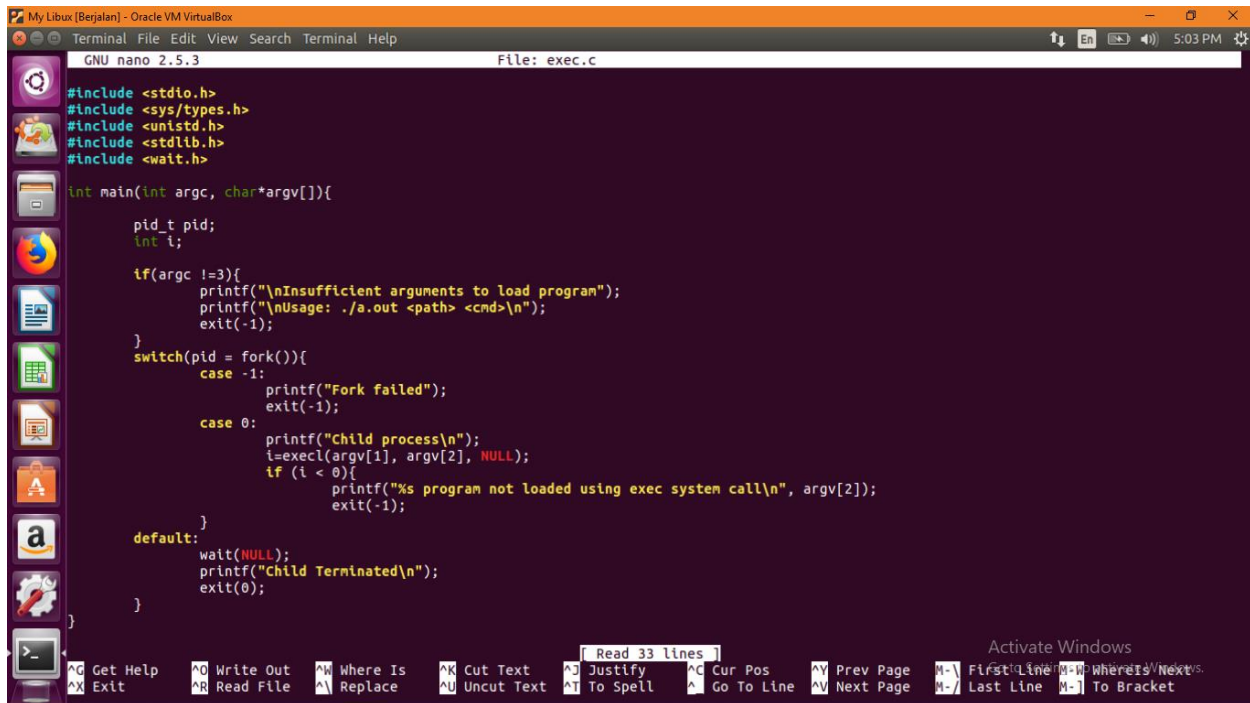
Activate Windows

Get Help Write Out Where Is Cut Text Justify Cur Pos Prev Page First Line Where is Next
Exit Read File Replace Uncut Text To Spell Go To Line Next Page Last Line To Bracket

```
ubuntu@ubuntu:~$ ./wait
Child starts
Nomor Ganjil: 1 3 5 7 9
Child ends

Parent starts
Nomor Genap: 2 4 6 8 10
Parent ends
ubuntu@ubuntu:~$ nano wait.c
ubuntu@ubuntu:~$
```

Exec.c



The screenshot shows a terminal window titled "My Linux [Berjalan] - Oracle VM VirtualBox". The terminal is running the GNU nano 2.5.3 editor, editing a file named "exec.c". The code is as follows:

```
#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.h>
#include <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(0);
    }
}
```

The terminal window has a menu bar with options: Terminal, File, Edit, View, Search, Terminal, Help. The status bar at the bottom shows "Read 33 lines" and "Activate Windows".

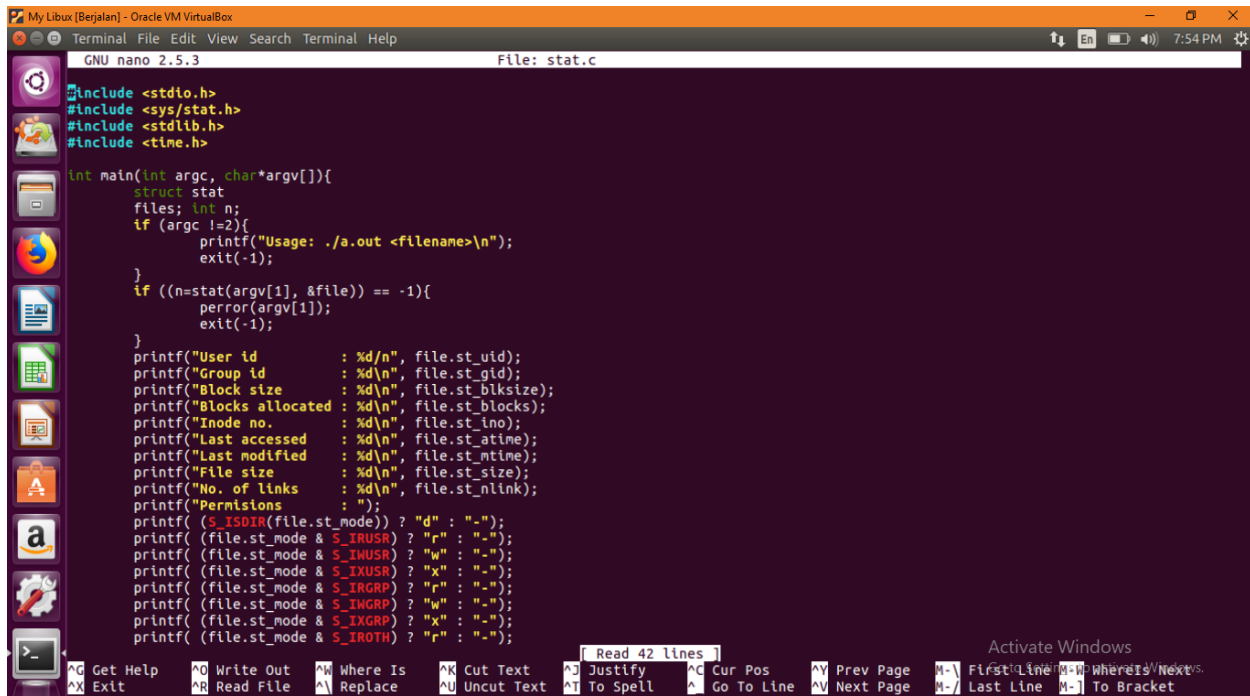


The screenshot shows a terminal window with the following commands and output:

```
ubuntu@ubuntu:~$ nano exec.c
ubuntu@ubuntu:~$ gcc exec.c -o exec
ubuntu@ubuntu:~$ nano exec.c
Amazon :~$ gcc exec.c -o exec
ubuntu@ubuntu:~$ ./exec
Insufficient arguments to load program
Usage: ./a.out <path> <cmd>
ubuntu@ubuntu:~$ ./a.out /bin/is is
bash: ./a.out: No such file or directory
ubuntu@ubuntu:~$ ./a.out
bash: ./a.out: No such file or directory
ubuntu@ubuntu:~$
```

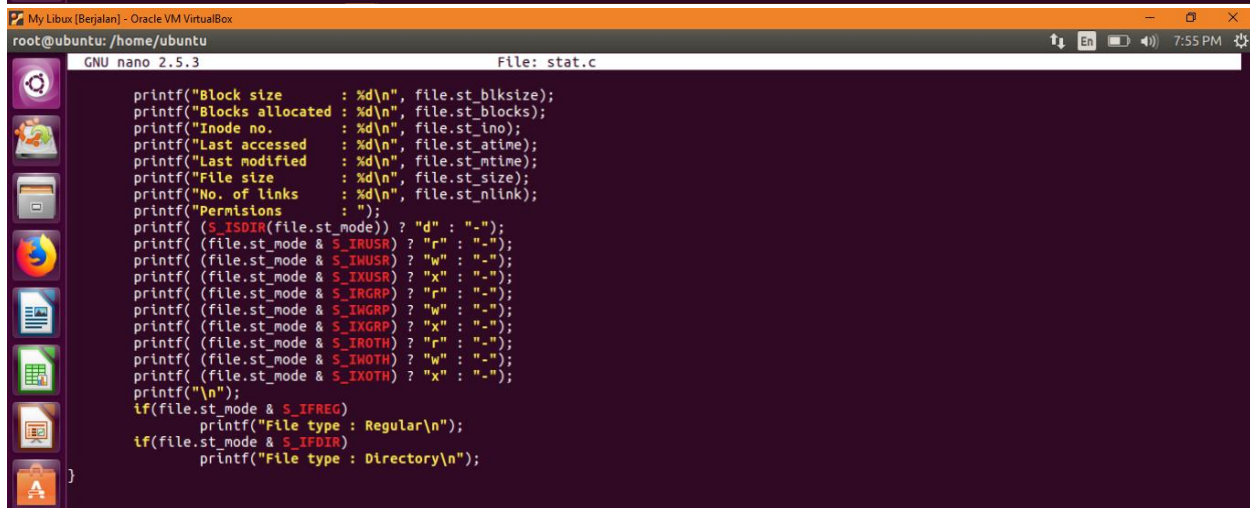
The terminal window has a menu bar with options: Get Help, Write Out, Where Is, Cut Text, Justify, Cur Pos, Prev Page, First Line, Where Is, Next. The status bar at the bottom shows "Activate Windows" and "Go to Settings to activate Windows."

Stat.c

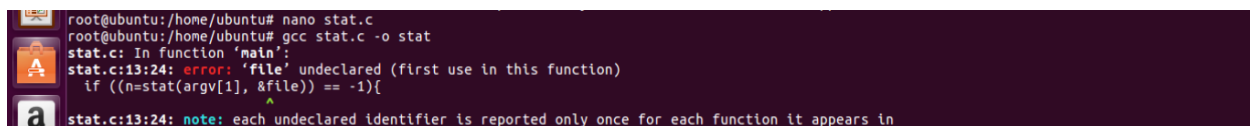


```
GNU nano 2.5.3 File: stat.c
#include <stdio.h>
#include <sys/stat.h>
#include <stdlib.h>
#include <time.h>

int main(int argc, char*argv[]){
    struct stat
    files; 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("Group id       : %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   : %d\n", file.st_atime);
    printf("Last modified   : %d\n", file.st_mtime);
    printf("File size       : %d\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");
}
```

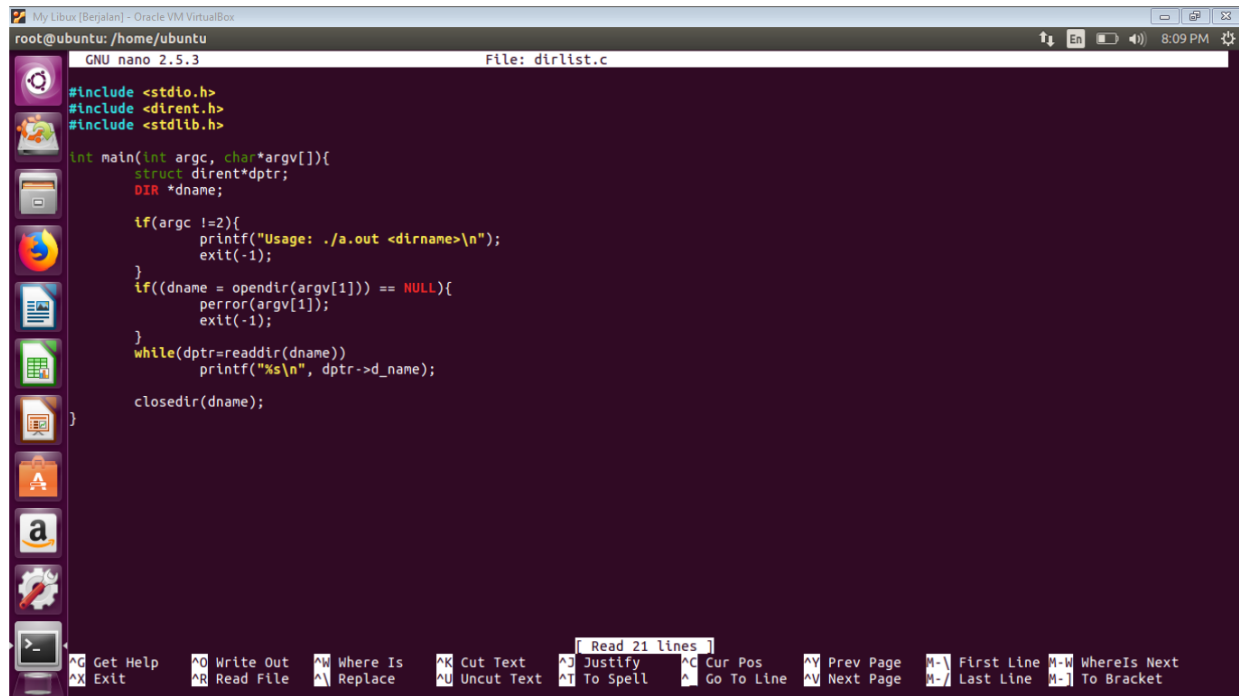


```
GNU nano 2.5.3 File: stat.c
printf("Block size      : %d\n", file.st_blksize);
printf("Blocks allocated : %d\n", file.st_blocks);
printf("Inode no.       : %d\n", file.st_ino);
printf("Last accessed   : %d\n", file.st_atime);
printf("Last modified   : %d\n", file.st_mtime);
printf("File size       : %d\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");
}
```



```
root@ubuntu:/home/ubuntu# nano stat.c
root@ubuntu:/home/ubuntu# gcc stat.c -o stat
stat.c: In function 'main':
stat.c:13:24: error: 'file' undeclared (first use in this function)
    if ((n=stat(argv[1], &file)) == -1){
                        ^
stat.c:13:24: note: each undeclared identifier is reported only once for each function it appears in
```

Dirlist.c



The screenshot shows a terminal window with the nano text editor open. The file being edited is `dirlist.c`. The code is as follows:

```
#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);
}
```

The nano editor's status bar at the bottom shows "Read 21 lines" and various keyboard shortcuts for editing and navigation.



The screenshot shows a terminal window with the following commands and output:

```
root@ubuntu:/home/ubuntu# nano dirlist.c
root@ubuntu:/home/ubuntu# gcc dirlist.c -o dirlist
root@ubuntu:/home/ubuntu# ./dirlist
Usage: ./a.out <dirname>
root@ubuntu:/home/ubuntu#
```

On the right side of the terminal window, there is a watermark that says "Activate Windows" and "Go to Settings to activate Windows."