

Nama : Riyan Aldiansyah  
NIM : L200170018  
Kelas : B

## MODUL 8

The image shows a Linux environment on a Lenovo K6 NOTE. A file manager window displays a directory with a file named 'wait.c'. A terminal window shows the execution of a C program that demonstrates process creation and management. The program uses `fork()` to create child processes, `wait()` to wait for child processes to complete, and `waitpid()` to wait for a specific child process. The output shows the parent process creating child processes, the child processes executing, and the parent process waiting for them to complete.

```
rialdi@rialdi-HP-14-Notebook-PC: ~/L200170018
File Edit View Search Terminal Help

rialdi@rialdi-HP-14-Notebook-PC:~/L200170018$ gcc -o fork fork.c
rialdi@rialdi-HP-14-Notebook-PC:~/L200170018$ ./fork

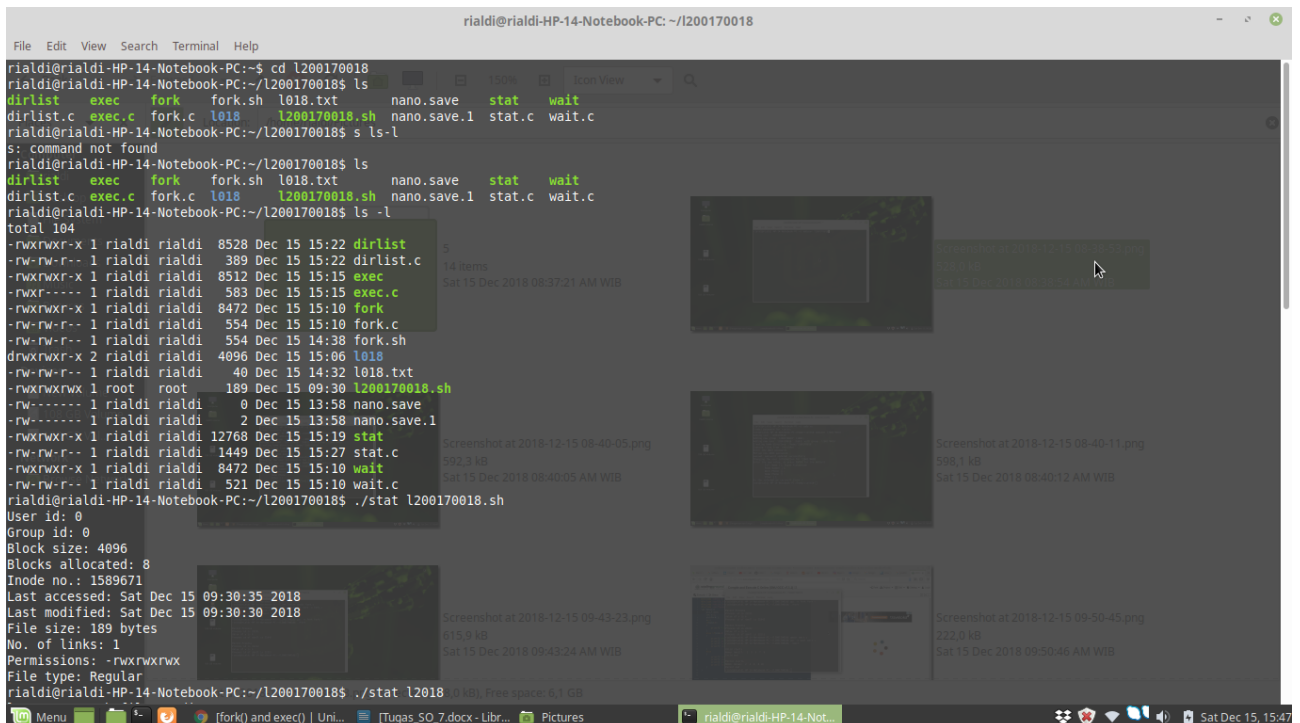
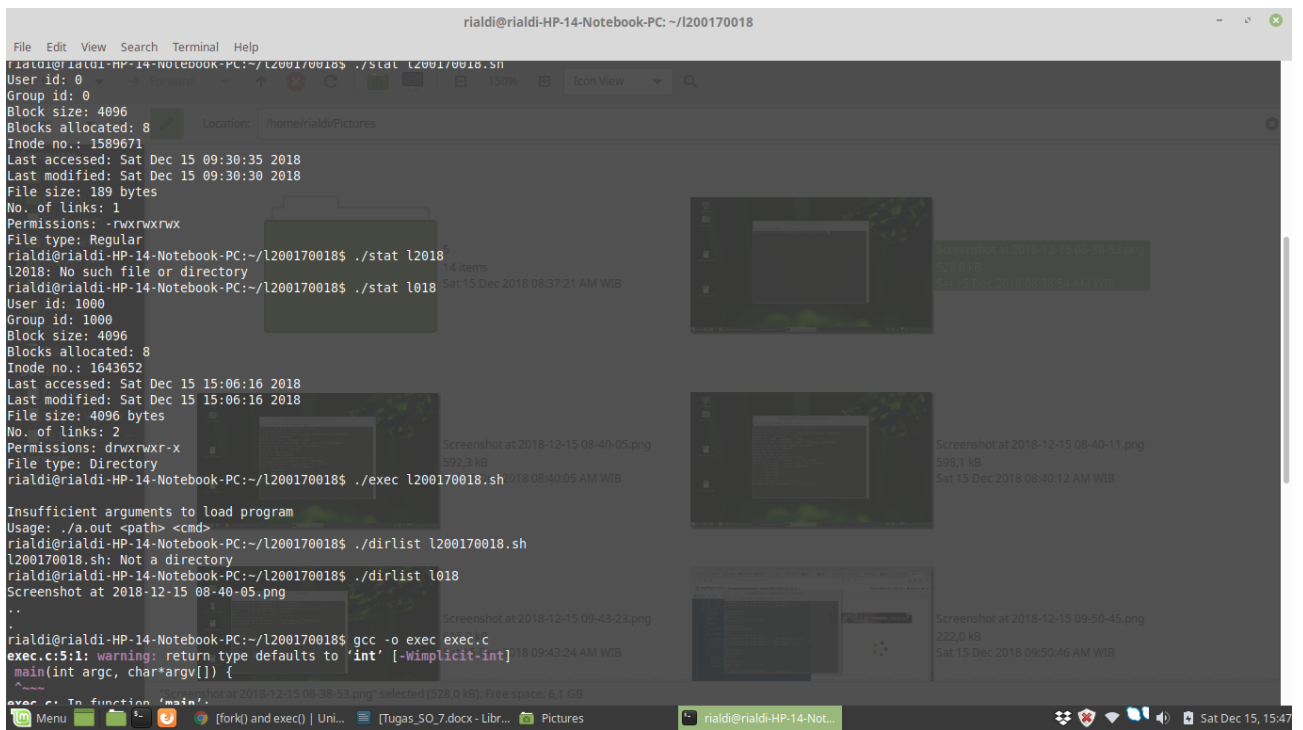
Parent Process
Process id is 22401
Value of x is 6
Process id of shell is 22384
Child process
Process id is 22402
Value of x is 6
Process id of parent is 22401

rialdi@rialdi-HP-14-Notebook-PC:~/L200170018$ gedit wait.c
rialdi@rialdi-HP-14-Notebook-PC:~/L200170018$ gcc -o wait wait.c
rialdi@rialdi-HP-14-Notebook-PC:~/L200170018$ ./wait

Child starts
Nomor Ganjil: 1 3 5 7 9
Child end

Parent starts
Nomor Genap: 2 4 6 8 10
Parent end
rialdi@rialdi-HP-14-Notebook-PC:~/L200170018$
```

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <unistd.h>
4 #include <sys/types.h>
5 int main()
6 {
7     pid_t pid;
8     int x=5;
9     pid=fork();
10    x++;
11    if(pid==0)
12    {
13        printf("Parent Process\n");
14        printf("Process id is %d\n", getpid());
15        printf("Value of x is %d\n", x);
16        printf("Process id of shell is %d\n", getpid());
17        printf("Child process\n");
18        printf("Process id is %d\n", getpid());
19        printf("Value of x is %d\n", x);
20        printf("Process id of parent is %d\n", getppid());
21    }
22    else
23    {
24        printf("Parent starts\n");
25        printf("Nomor Ganjil: ");
26        for(i=1; i<10; i+=2)
27        {
28            printf("%d ", i);
29        }
30        printf("\n");
31        printf("Child end\n");
32        printf("Parent ends\n");
33        printf("Nomor Genap: ");
34        for(i=2; i<10; i+=2)
35        {
36            printf("%d ", i);
37        }
38        printf("\n");
39    }
40    return 0;
41 }
```



```
rialdi@rialdi-HP-14-Notebook-PC: ~/l200170018
File Edit View Search Terminal Help

rialdi@rialdi-HP-14-Notebook-PC:~/l200170018$ gcc -o exec exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char*argv[]) {
^
rialdi@rialdi-HP-14-Notebook-PC:~/l200170018$ ./a.out
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);
    ^
rialdi@rialdi-HP-14-Notebook-PC:~/l200170018$ ./a.out
bash: ./a.out: No such file or directory
rialdi@rialdi-HP-14-Notebook-PC:~/l200170018$ gcc exec.c
exec.c:5:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(int argc, char*argv[]) {
^
rialdi@rialdi-HP-14-Notebook-PC:~/l200170018$ ./a.out
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);
    ^
rialdi@rialdi-HP-14-Notebook-PC:~/l200170018$ ./a.out
Insufficient arguments to load program
Usage: ./a.out <path> <cmd>
rialdi@rialdi-HP-14-Notebook-PC:~/l200170018$ ./exec.c
bash: ./exec.c: Permission denied
rialdi@rialdi-HP-14-Notebook-PC:~/l200170018$ ./fork.c
bash: ./fork.c: Permission denied
rialdi@rialdi-HP-14-Notebook-PC:~/l200170018$
```