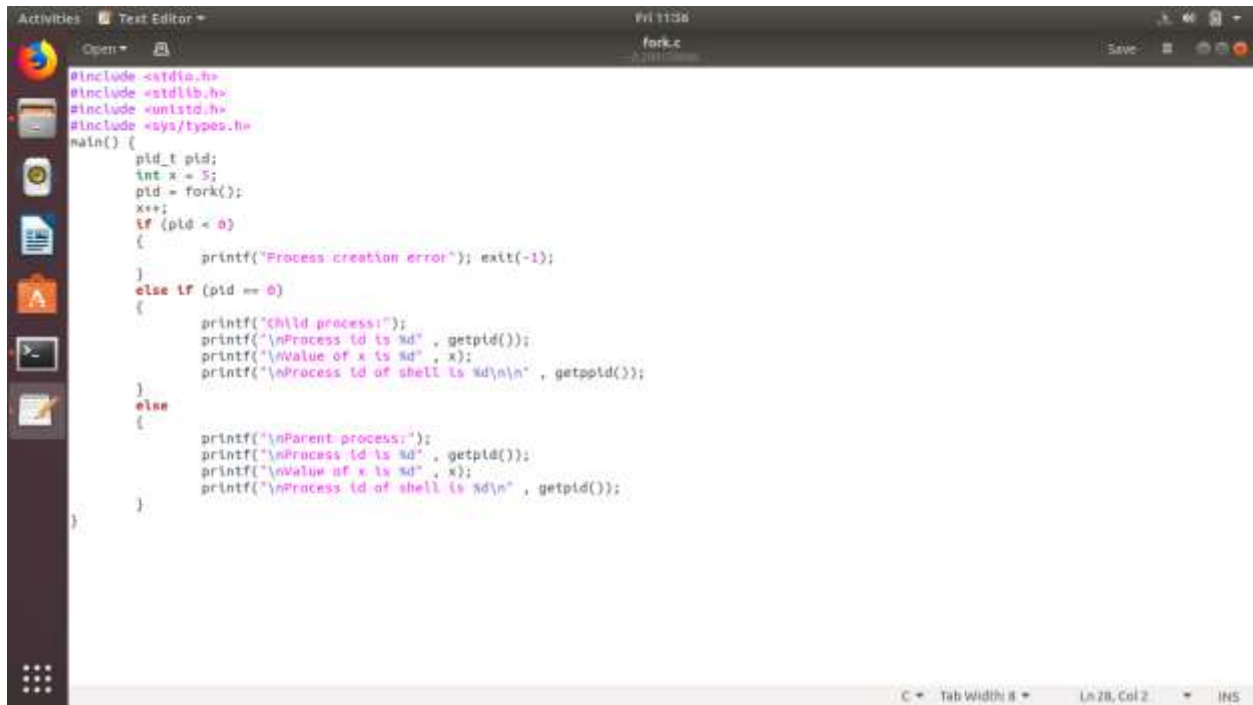


NAMA : TIKA PRATIWI

NIM : L200170046

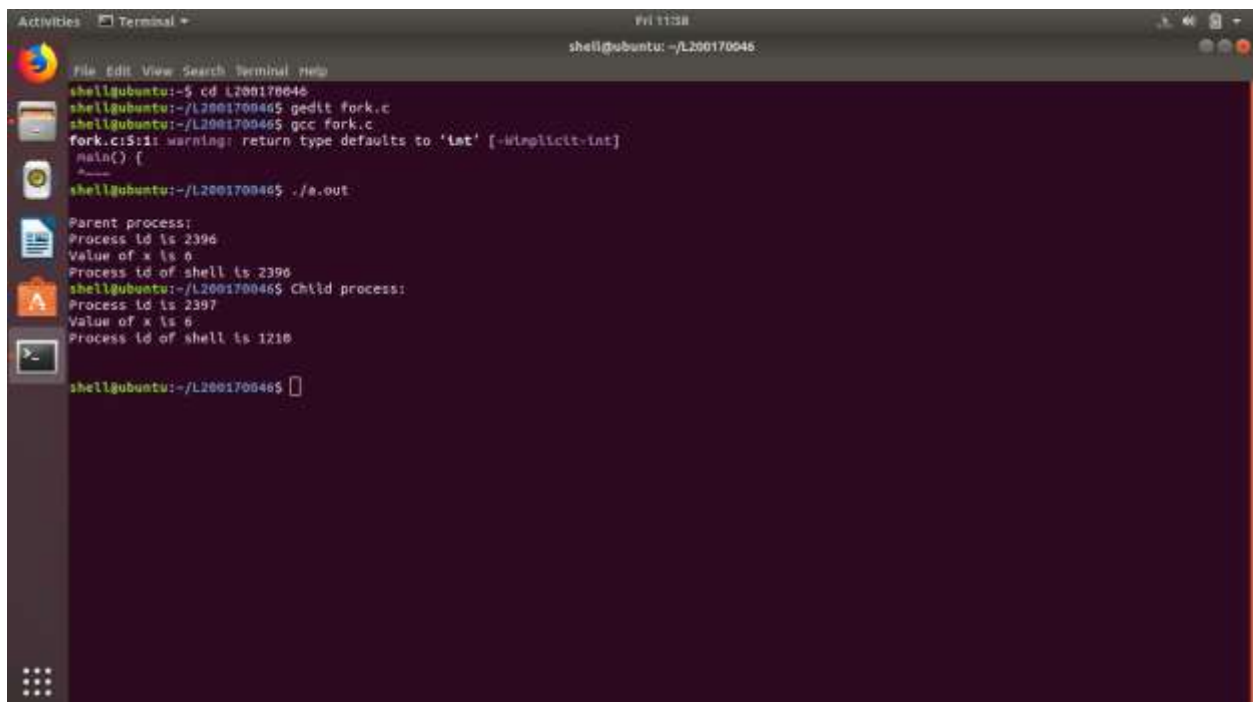
KELAS : C

## MODUL 8



```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>

main() {
    pid_t pld;
    int x = 5;
    pld = fork();
    x++;
    if (pld < 0)
    {
        printf("Process creation error"); exit(-1);
    }
    else if (pld == 0)
    {
        printf("Child process:");
        printf("\nProcess id is %d", getpid());
        printf("\nValue of x is %d", x);
        printf("\nProcess id of shell 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());
    }
}
```



```
shell@ubuntu:~$ cd /L200170046
shell@ubuntu:~/L200170046$ gedit fork.c
shell@ubuntu:~/L200170046$ gcc fork.c
fork.c:5:11: warning: return type defaults to 'int' [-Wimplicit-int]
    main() {
    ^~~~~
shell@ubuntu:~/L200170046$ ./a.out
Parent process:
Process id is 2396
Value of x is 6
Process id of shell is 2396
shell@ubuntu:~/L200170046$ Child process:
Process id is 2397
Value of x is 6
Process id of shell is 1210
shell@ubuntu:~/L200170046$
```

```
Activities Text Editor + Fri 12:01
wait.c
Save

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>
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(" %d", i);
        printf("\nParent ends\n");
    }
    else if (pid == 0)
    {
        printf("\nChild starts\nNomor Ganjil:");
        for (i=1; i<=10; i+=1)
            printf(" %d", i);
        printf("\nChild ends\n");
    }
}
```

Saving file "/home/shell/L200170046/wait.c"...

C Tab Width: 8 Ln 30, Col 2 INS

```
Activities Terminal + Fri 12:02
shell@ubuntu: ~/L200170046
File Edit View Search Terminal Help

shell@ubuntu:~$ cd L200170046
shell@ubuntu:~/L200170046$ gedit wait.c
shell@ubuntu:~/L200170046$ gcc wait.c
wait.c:6:11: warning: return type defaults to 'int' [-Wimplicit-int]
    main() {
    ^~~~~~

shell@ubuntu:~/L200170046$ ./a.out
Child starts
Nomor Ganjil: 1 3 5 7 9
Child ends

Parent starts
Nomor Genap: 2 4 6 8 10
Parent ends
shell@ubuntu:~/L200170046$
```

```
Activities Text Editor + Fri 12:00
EXEC.C
Save

#include <stdio.h>
#include <sys/types.h>
#include <unistd.h>
#include <stdlib.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], 0);
            if (i < 0)
            {
                printf("%s program not loaded using exec system call\n", argv[2]);
                exit(-1);
            }
        default:
            printf("Child Terminated\n");
            exit(0);
    }
}
```

Saving file "/home/shell/L200170046/exec.c"...

C Tab Width: 8 Ln 34, Col 2 INS

```
Activities Terminal + Fri 12:05
shell@ubuntu: ~/L200170046
File Edit View Search Terminal Help

shell@ubuntu:~/L200170046$ ./L200170046
L200170046: command not found
shell@ubuntu:~/L200170046$ cd L200170046
shell@ubuntu:~/L200170046$ gedit exec.c
shell@ubuntu:~/L200170046$ gcc exec.c
exec.c: in function 'main':
exec.c:23:3: warning: missing sentinel in function call [-Wformat=]
    i = execl(argv[1], argv[2], 0);
    ^
shell@ubuntu:~/L200170046$ ./a.out
Insufficient arguments to load program
Usage: ./a.out <path> <cmd>
shell@ubuntu:~/L200170046$ ./a.out /bin/ls /ls
Child Terminated
shell@ubuntu:~/L200170046$ Child process
a.out exec.c fork.c wait.c
shell@ubuntu:~/L200170046$
```

Activities Text Editor + Fri 12:07

Open Save

```
#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);
}
```

Saving file "/home/shell/L200170046/dirlist.c"...

C Tab Width: 8 Ln 21, Col 2 INS

Activities Terminal + Fri 12:11

File Edit View Search Terminal Help

shell@ubuntu: ~/L200170046

```
shell@ubuntu:~$ cd L200170046
shell@ubuntu:~/L200170046$ gedit dirlist.c
shell@ubuntu:~/L200170046$ gcc dirlist.c
shell@ubuntu:~/L200170046$ ./a.out .
a.out
exec.c
dirlist.c
wait.c
fork.c
..
shell@ubuntu:~/L200170046$
```