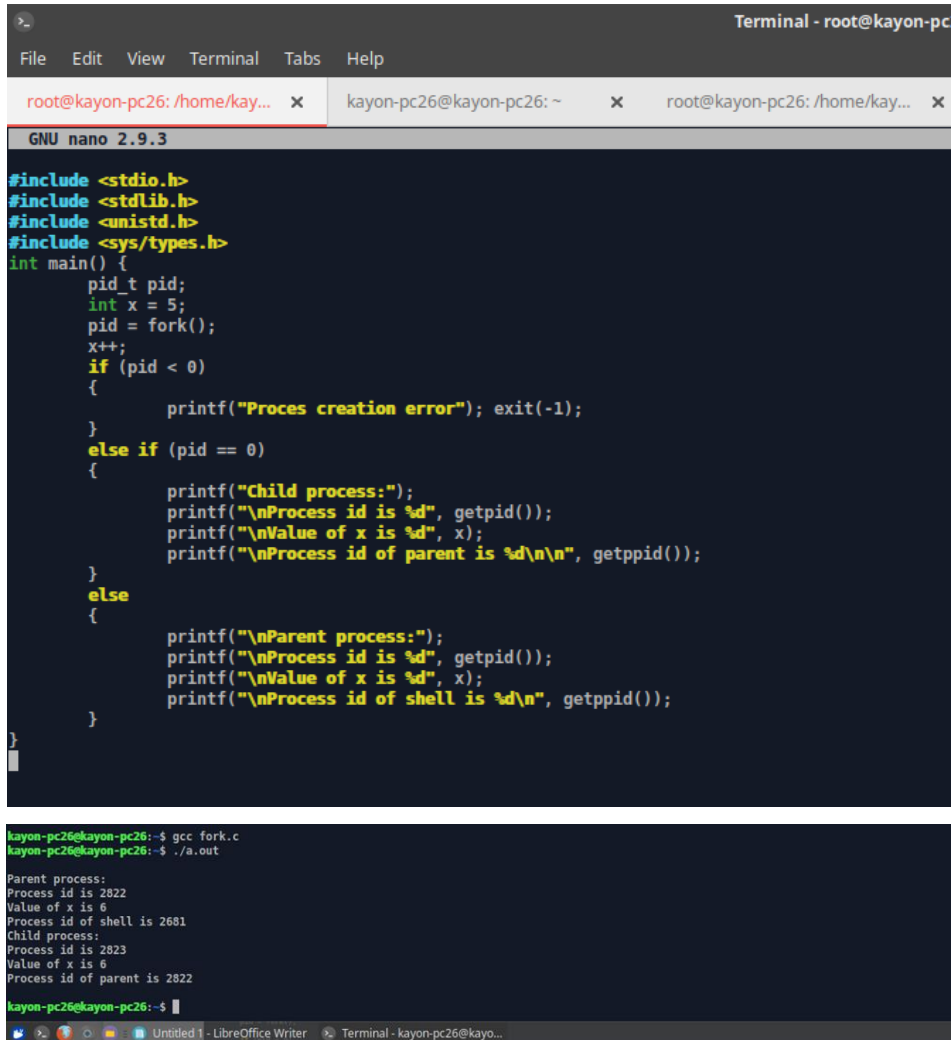


NAMA : DHIYA ULHAQ A

NIM : L200180009

MODUL 8



```
Terminal - root@kayon-pc26: ~
File Edit View Terminal Tabs Help
root@kayon-pc26: /home/kay... x kayon-pc26@kayon-pc26: ~ x root@kayon-pc26: /home/kay... x
GNU nano 2.9.3
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/types.h>
int main() {
    pid_t pid;
    int x = 5;
    pid = fork();
    x++;
    if (pid < 0)
    {
        printf("Proces 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\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());
    }
}

kayon-pc26@kayon-pc26:~$ gcc fork.c
kayon-pc26@kayon-pc26:~$ ./a.out
Parent process:
Process id is 2822
Value of x is 5
Process id of shell is 2681
Child process:
Process id is 2823
Value of x is 6
Process id of parent is 2822
kayon-pc26@kayon-pc26:~$
```

```
Terminal - root@kayon-pc26
File Edit View Terminal Tabs Help
root@kayon-pc26: /home/kay... x kayon-pc26@kayon-pc26: ~ x root@kayon-pc26: /home/kay... x
GNU nano 2.9.3
#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");
    }
}

kayon-pc26@kayon-pc26:~$ gcc wait.c
kayon-pc26@kayon-pc26:~$ ./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-pc26@kayon-pc26:~$
```

```
Terminal - root@kayon-pc26: /home/kayon
File Edit View Terminal Tabs Help
root@kayon-pc26: /home/kay... x kayon-pc26@kayon-pc26: ~ x root@kayon-pc26: /home/kay... x kayon-pc26@
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(0);
    }
}

kayon-pc26@kayon-pc26:~$ gcc exec.c
kayon-pc26@kayon-pc26:~$ ./a.out /bin/ls ls
Child process
Child Terminated
kayon-pc26@kayon-pc26:~$
```

Activities Terminal ▼ Kam 11:46

root@panda-VirtualBox: /home/panda

File Edit View Search Terminal Tabs Help

panda@pan... x root@pand... x panda@pan... x panda@pan... x root@pand... x

GNU nano 2.9.3 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("Group id : %d\n", file.st_gid);
    printf("Block size : %ld\n", file.st_blksize);
    printf("Blocks allocated : %ld\n", file.st_blocks);
    printf("Inode no. : %ld\n", file.st_ino);
    printf("Last accessed : %s", ctime(&(file.st_atime)));
    printf("Last modified : %s", 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_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");
}
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell

Activities Terminal Kam 11:48

panda@panda-VirtualBox: ~

File Edit View Search Terminal Tabs Help

panda@pan... x root@pand... x panda@pan... x panda@pan... x root@pand... x

```
panda@panda-VirtualBox:~$ gcc stat.c
panda@panda-VirtualBox:~$ ./a.out stat.c
User id : 0
Group id : 0
Block size : 4096
Blocks allocated : 8
Inode no. : 131874
Last accessed : Thu Nov 21 11:36:57 2019
Last modified : Thu Nov 21 11:36:45 2019
File size : 1561 bytes
No. of links : 1
Permissions : -rw-r--r--
File type : Regular
panda@panda-VirtualBox:~$
```

Activities Terminal Kam 11:49

root@panda-VirtualBox: /home/panda

File Edit View Search Terminal Tabs Help

panda@pan... x root@pand... x panda@pan... x panda@pan... x root@pand... x

GNU nano 2.9.3 dirlist.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);
}
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell

