**LABORATOR11**

**Clientul ii transmite serverului un un nume de director, iar serverul ii retransmite clientului numarul total de bytes din toate fisierele din directorul respectiv.**

#include<stdlib.h>

#include<unistd.h>

#include<string.h>

#include<wait.h>

#include<dirent.h>

#include<sys/wait.h>

#include<stdio.h>

#include<sys/types.h>

#include<sys/stat.h>

#include<errno.h>

#include<fcntl.h>

extern int errno;

int main()

{

int f1,f2;

int pid;

char ser[50];

char sir1[100];

if(mkfifo("fifo1", 0666) == -1 )

{

if(errno==17) //17= errno for "file exists"

{

fprintf(stdout,"Note: fifo %s exista deja!\n","fifo1");

exit(0);

}

else

{

fprintf(stderr,"Error: creare fifo imposibila, errno=%d\n", errno);

perror(0);

exit(2);

}

}

if( mkfifo("fifo2", 0666) == -1)

{

if(errno==17) //17=errno for "File exists"

{

fprintf(stdout,"Note: fifo %s exista deja!\n", "fifo2");

exit(0);

}

else

{

fprintf(stderr,"Error: create fifo imposible, errno=%d\n",errno);

perror(0);

exit(2);

}

}

pid=fork();

if(pid<0)

{

printf("eroare la fork\n");

exit(1);

}

if(pid==0)

{

//SERVERUL=procesul fiu

f1=open("fifo1", O\_RDONLY);

read(f1,ser,50);

printf("SERVERUL\nNumele de director primit este:%s\n",ser);

char \*cmd=(char\*)malloc(sizeof(char)\*30);

strcpy(cmd,"du ");

strcat(cmd,ser);

FILE \*f=popen(cmd,"r");

int r;

fscanf(f,"%d",&r);

pclose(f);

f2=open("fifo2", O\_WRONLY);

write(f2,&r,100);

close(f1);

close(f2);

}

else

{

//CLIENTUL

f1=open("fifo1", O\_WRONLY);

printf("CLIENTUL:\nDati un nume de director:");

scanf("%s",ser);

write(f1,ser,(strlen(ser)+1));

f2=open("fifo2", O\_RDONLY);

int ra;

read(f2,&ra,100);

printf("CLIENTUL:\n Dimensiunea directorului este: %d\n",ra);

close(f1);

close(f2);

}

unlink("fifo1");

unlink("fifo2");

//return(0);

}