#include <stdio.h>

#include <stdlib.h>

#include <unistd.h>

#include <string.h>

#include <errno.h>

int main(int argc, char \*argv[ ])

{

int i; /\* numero de este proceso (comenzando en 1) \*/

int childpid; /\* para recibir el resultado de fork \*/

int nprocs; /\* nº total de procesos en el anillo \*/

int fd[2];

int num;

int error;

int valor;

if ( (argc!= 2) || ((nprocs=atoi(argv[1])) <= 0) ) {

fprintf(stderr, "Error en %s Debe dar como argumento el nº de procesos nprocs\n",argv[0]);

exit(1); }

if (pipe (fd) == -1) {

perror("No se puede crear pipe");

exit(1);

}

if ((dup2(fd[0], STDIN\_FILENO) == -1) || (dup2(fd[1], STDOUT\_FILENO) == -1))

{

perror("Could not dup pipes");

exit(1);

}

if ((close(fd[0]) == -1) || (close(fd[1]) == -1)) { //AL DUPLICAR CON EL DUP2 cerramos los fd y las salidas STDIN Y STDOUT seran ahora para el cauce OJO!!

perror("Could not close extra descriptors"); //DA IGUAL SI LOS CIERRO O NO, PK FUNCIONARA ESCRIBAS en los fd o STDIN,STDOUT

exit(1);

}

valor=0;///////////////////////////////////////////

if(write(STDOUT\_FILENO,&valor,sizeof(int))<0)

{

perror("..");

exit(-1);

}

/////////////////////////////////////////////////

// crea los restantes procesos y sus conexiones

for (i = 1; i <nprocs; i++) {

if (pipe (fd) == -1) { //al ser un anillo crea piped para conectar con uno y otro

fprintf(stderr,"Could not create pipe %d: %s\n",

i,strerror(errno));

exit(1);

}

if ((childpid= fork()) == -1) { //cerramos el hijo

fprintf(stderr, "Could not create child %d: %s\n",

i,strerror(errno));

exit(1);

}

if (childpid> 0) //esto lo hace el padre

error = dup2(fd[1], STDOUT\_FILENO);//ES PARA ENLAZAR HIJO

else //hijo

error = dup2(fd[0], STDIN\_FILENO);

if (error == -1) {

fprintf(stderr, "Could not dup pipes for iteration %d: %s\n",i,strerror(errno));

exit(1);

}

if ((close(fd[0]) == -1) || (close(fd[1]) == -1)) {

fprintf(stderr, "Could not close extra descriptors %d: %s\n",i,strerror(errno));

exit(1);

}

if (childpid) //EL PADRE SE SALE Y EL HIJO CREA OTRO HIJO

{

break;

}

}// for

while(valor!=9 && read(STDIN\_FILENO,&valor,sizeof(int)))

{

fprintf(stderr,"Proceso con i= %d, pid %d y valor %d\n",i, (int)getpid(), valor);

if(valor!=9)

valor++;

write(STDOUT\_FILENO,&valor,sizeof(int));

}

//fprintf(stderr,"Proceso con i= %d, pid %d y padre %d\n",i, (int)getpid(), (int)getppid());

exit(0);

}