PRACTICA 8

Created	@January 15, 2021 10:31 AM
Date	@November 30, 2020 → December 13, 2020
Tags	
Tema	

Tema i què es fa

Gestión entrada/salida. Se tratan las pipes. Pipes sin nombre, con nombre y sockets. Comunicamos procesos con pipes y redirigimos su salida y entrada a pipes con y sin nombre.

Sessió i codis

▼ escritor.c

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

```
int main(int argc, char **argv){
  char buff[80];
  int fd = open("./pipe",O_WRONLY);
  int ret;
  while((ret = read(fd,buff,ret)) > 0){
    write(fd,buff,ret);
  }
}
```

▼ escritor_socket.c

```
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include "socketMng.c"

int main(int argc, char **argv){
  int socketFD;
  int connectionFD;
  int ret;
  char buffer[80];

  if (argc != 2) {
    strcpy(buffer, "Usage: prServerSocket socketPath\n");
    write(2, buffer, strlen(buffer));
    exit(1);
  }

  socketFD = createSocket(argv[1]);
  connectionFD = serverConnection(socketFD);
  ret = read(0, buffer, sizeof(buffer));

  while (ret > 0) {
    write(connectionFD, buffer, ret);
    ret = read(0, buffer, sizeof(buffer));
  }

  closeConnection(connectionFD);
  deleteSocket(socketFD, argv[1]);

}
```

▼ escritor_v2.c

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

```
#include <errno.h>

int main(int argc, char **argv){
  char buff[80];
  int fd = open("./pipe",O_WRONLY|O_NONBLOCK);
  if ((fd < 0) && (errno == ENXIO)) perror("Esperando lector");
  fd = open("./pipe",O_WRONLY);
  int ret;
  while((ret = read(fd,buff,ret)) > 0){
    write(fd,buff,ret);
  }
}
```

▼ lector.c

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

int main(int argc, char **argv){
  char buff[80];
  int fd = open("./pipe",O_RDONLY);
  int ret;
  while((ret = read(fd,buff,ret)) > 0){
    write(1,buff,ret);
  }
}
```

▼ lector_socket.c

```
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include "socketMng.c"

int main(int argc, char **argv){
  int connectionFD;
  int ret;
  char buffer[80];

  if (argc != 2) {
    strcpy(buffer, "Usage: prClientSocket socketPath\n");
    write(2, buffer, strlen(buffer));
    exit(1);
  }

  connectionFD = clientConnection(argv[1]);
  ret = read(connectionFD, buffer, sizeof(buffer));
```

```
  while (ret > 0) {
    write(1, buffer, ret);
    ret = read(connectionFD, buffer, sizeof(buffer));
  }

  closeConnection(connectionFD);

}
```

▼ sin_nombre.c

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

main (int argc, char *argv[])
{
  char buff[80];
  int p[2];
  pipe(p);
  int pid = fork();
  if (pid == 0){
    //close(0);
    dup2(p[0],0);
    close(p[0]);
    close(p[1]);
    execlp("cat","cat",(char*)0);
  }
  else if(pid > 0){
    close(p[0]);
    sprintf(buff,"Inicio\n");
    write(p[1],buff,strlen(buff));
    close(p[1]);
    waitpid(-1,NULL,0);
    sprintf(buff,"Fin\n");
    write(1,buff,strlen(buff));
  }
}
```

Manual

❓ pipe - Crea una pipe sin nombre

❓ open - Abre un fichero

❓ dup/dup2 - Duplica un descriptor de fichero

❓ socket - Crea un socket

❓ bind - Asigna dirección a un socket

❓ connect - Conecta un socket