

## Tubos (Pipes)

Pipes são o mecanismo original de comunicação unidirecional entre processos no Unix que utiliza comunicação via núcleo do S.O, esse mecanismo é uma estrutura que possui duas extremidades, que são implementadas como arquivos (o processo pode escrever e ler destas extremidades), formando uma fila de caracteres do tipo FIFO.

Um pipe é caracteriza-se por ser:

- Anônimo (não tem nome).
- Temporário: dura somente o tempo de execução do processo que o criou (não permanece no sistema de arquivos).

Os pipes possuem capacidade limitada:

- Se uma escrita é feita e existe espaço no pipe, o dado é colocado no pipe e a chamada retorna imediatamente.
- Se a escrita sobre um pipe continua mesmo depois dele estar cheio, ocorre uma situação de bloqueio (que permanece até que algum outro processo leia e, consequentemente, abra espaço no pipe).
- Não existe nenhum mecanismo que permita diferenciar as informações na saída do pipe.

## Criação de Pipes

Os Pipes constituem um canal de comunicação entre processos pai-fillhos, sendo definidos antes da criação dos processos descendentes (podendo ligar apenas processos com antepassado comum).

Criado por uma chamada de sistema e são retornados dois descritores:

Chamada de sistema:

```
#include <unistd.h>
int pipe(int fd[2])
```

Descritores:

- Descritor fd[0] - aberto para leitura
- Descritor fd[1] - aberto para escrita.

Quando um processo faz um fork() depois de criado o pipe, o processo filho recebe os mesmos descritores de leitura e escrita do pai. Cada um dos processos deve fechar a extremidade não aproveitada do pipe.

Ex:

- se o pai vai escrever, ele fecha o descritor de leitura
- se o filho vai ler, ele fecha o descritor de escrita

## Fechamento de Pipes

Depois de usados, ambos os descritores devem ser fechados pela chamada do sistema, todos os dados do pipe são perdidos, em caso de erro retorna -1 com sua causa indicada na variável de ambiente `int errno` e em caso de sucesso, retorna 0.

Ex:

```
#include <unistd.h>
int close (int);
...
int fd[2];
if (pipe(fd)==0) {
    ...
    close(fd[0]); close(fd[1]);
}
```

## Escrita e Leitura em Pipes

A escrita e leitura é feita pela seguinte chamada de sistema:

```
#include <unistd.h>
ssize_t read(int, char *, int);
ssize_t write(int, char *, int);
```

- 1º parâmetro: descritor de arquivo.
- 2º parâmetro: endereço dos dados.
- 3º parâmetro: número de bytes a comunicar.