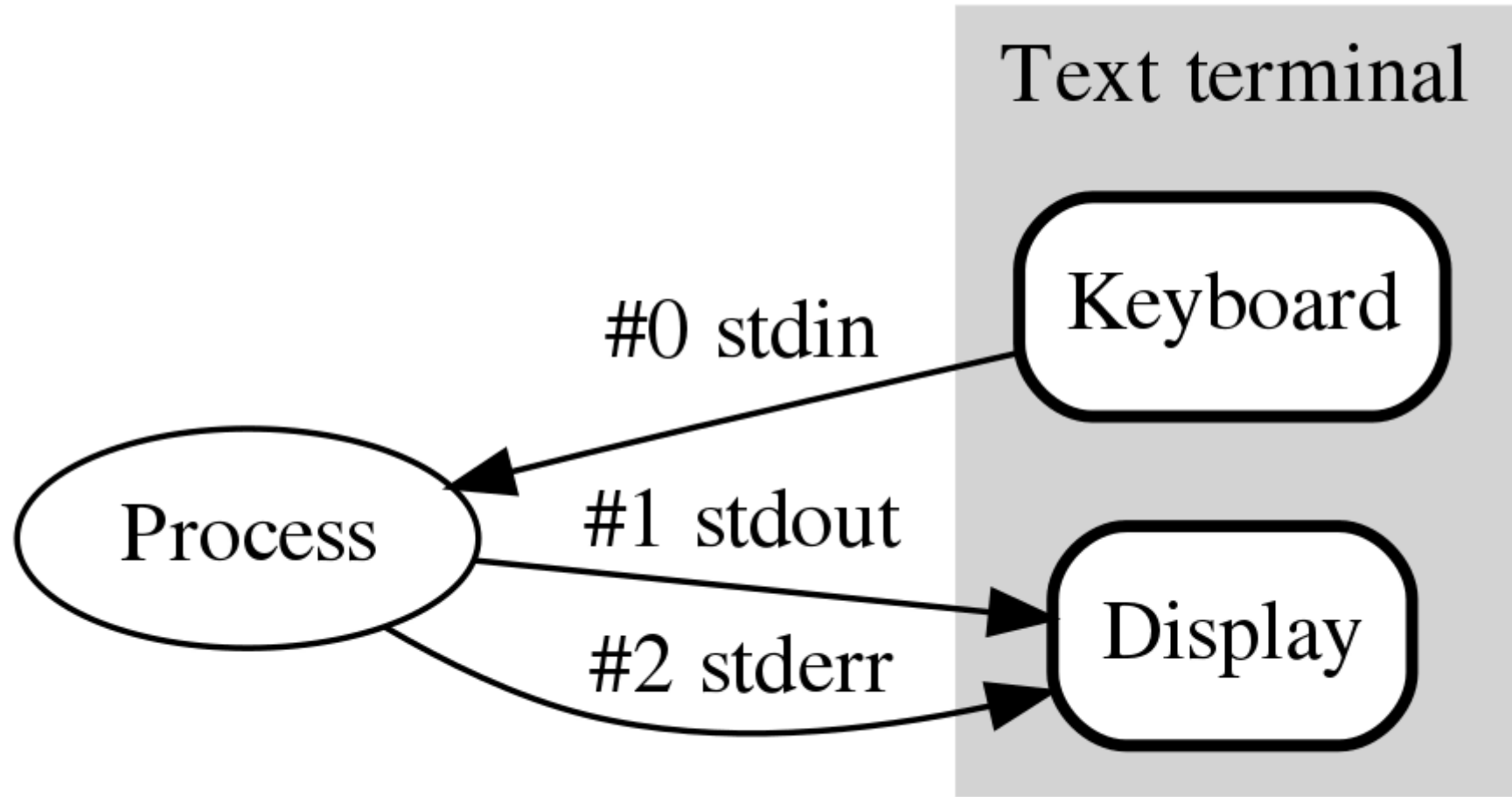


Fluxos de E/S Padrão e Operadores de Fluxo

Os fluxos padrão são canais de entrada/saída (**E/S**) entre um programa de computador e o seu ambiente (tipicamente um terminal de texto) que são pré conectados no início da execução.

- ✓ **ST**andar**D IN**put, ou Entrada Padrão - Todos os programas que fazem interface com o usuário precisam receber por algum meio as informações passadas por ele. O meio mais antigo e comum do usuário passar informações a um programa é via teclado. Ela pode ser representada pelo número **0**.
\$ **ls** /dev/stdin
- ✓ **ST**andar**D OUT**put, ou Saída Padrão - que representa o monitor, já que ele é o dispositivo de saída padrão na interface com o usuário. Nesta saída, temos acesso a todas as mensagens de informação que o sistema gera na tela para o usuário. Ela pode ser representada pelo número **1**.
\$ **ls** /dev/stdout
- ✓ **ST**andar**D ERR**or, ou Erro Padrão - é apresentado no monitor, mas é por ele que são enviadas as mensagens de erro geradas pelos aplicativos. Ela pode ser representada pelo número **2**.
\$ **ls** /dev/stderr



OPERADORES DE FLUXO

Você pode manipular as entradas e saídas com estes três operadores

- **Pipe/Pipeline (|)**: Liga o *stdout* de um programa ao *stdin* de outro.
- **Write (>)**: Redireciona o *stdout* para outro local (um arquivo, por exemplo).
- **Append (>>)**: Anexa o *stdout* para outro local (um arquivo, por exemplo).

Repare que há uma pequena diferença entre o “>” e o “>>”: o primeiro apaga o conteúdo do destino, para então escrever seus dados; o segundo apenas acrescenta as informações às já existentes.

Exemplos de uso: terminal

- ▶ Enviar a saída do comando **ls** para o arquivo “*lista.txt*” , se houver conteúdo em *lista.txt* será substituído pelo conteúdo do **ls**: `$ ls > lista.txt`
- ▶ Semelhante ao comando acima, mas preservando o conteúdo original do arquivo “*lista.txt*”, o conteúdo vai para o final do arquivo *lista.txt*: `$ ls >> lista.txt`
- ▶ lista os arquivos e os coloca em ordem alfabética: `$ ls | sort`

Vamos ver exemplos mais avançados de utilização desses operadores na segunda parte desse curso, tais como:

EOF, **HERE DOCUMENT**, **IFS**, ...