# Redirecionamento de entrada/saída padrão em sistemas POSIX

Aula teste para disciplina Sistemas de Hardware-Software



#### Agenda

- Assuntos anteriores
- Objetivos de aula
- Dinâmica da aula
- Entrada/Saída padrão de processos (Hands-on)
- Leitura e Escrita em arquivos (Hands-on)
- "File Descriptor", "Open File Table" e "iNode Table"
- "Tudo são arquivos"
- Redirecionamento de Entrada/Saída via código (Hands-on)
- Redirecionamento de Entrada/Saída via terminal (Hands-on)



#### Assuntos anteriores

#### O que sabemos?

- Linux
  - Trabalhar com terminal no Linux
    - Manipulação de pastas e arquivos
    - Navegação e estrutura básica das pastas
    - Encadeamento de processos
  - O que são processos
- Programação C
  - Variáveis, estrutura de dados
  - SystemCall
    - exec\*
    - open, close, read e write
  - Entrada/Saída padrão ("Teclado e Tela do Terminal")
    - scanf, getchar, gets (Entrada)
    - Printf, puts (Saída)
  - Criar/Clonar processos (fork)



#### Objetivos da aula

#### Objetivo Primário

 Compreender o funcionamento dos mecanismos de redirecionamento de entrada/saída padrão em sistemas POSIX

#### Objetivos Secundários

- Compreender o que é um File Descriptor
- Utilizar redirecionamento via terminal e via código
- Utilizar dup e dup2
- Compreender o que são sistemas POSIX



#### Dinâmica de Aula

- Apresentação
- Código e Terminal
- Participativa (Muito!)



#### Linux Terminal













#### Apresentação

Redirecionamento de entrada/saída padrão em sistemas POSIX



 ${\it Land idato: Prof. Tiago Sanches da Silva} \qquad {\it Insper}$ 



#### Exercício 1

Na pasta "1. exercicio-busca-arquivo" existe 3 pastas com algumas subpastas.

- 1.0 Gostaria de relembrar alguns comandos do terminal explorando estes arquivos;
- 1.1 Encontrar os arquivos que possuam a palavra "Insper" em seus conteúdos
- 1.2 Utilizar <u>lshw</u> para exibir informações sobre o hardware
  - 1.2.1 Listar apenas os nomes lógicos
  - 1.2.2 Salvar os nomes lógicos em um arquivo .txt



#### Pipe e Redirection (terminal)

Pipes "|": O Pipe permite juntar dois ou mais comandos executados em sequência. Passa a saída para um outro programa ou utilitário.

Redirecionamentos de saída (">" e ">>"): Redireciona a saída para um arquivo ou streams (arquivos, teclado, tela, sockets buffer etc)

- ">": Cria arquivo
- ">>": Adiciona ao final do arquivo

Redirecionamentos de entrada ("<"): O comando em vez de ler a informação de um teclado, ele lê a partir de um arquivo.





### E/S Padrão (Standard I/O)

O Linux utiliza três *streams* de E/S padrão, cada um associado a um descritor de arquivo conhecido:

- Entrada Padrão (stdin): A Entrada Padrão é um stream (fluxo) para entrada de texto, vinculada ao teclado. Descritor de arquivo (File Descriptor) 0
- Saída Padrão (stdout): A Saída Padrão é um stream (fluxo) para saída dos processos. Descritor de arquivo (File Descriptor) 1
- Erro Padrão (stderr): O Erro Padrão também é um stream de saída de texto, porém usado para exibir mensagens de erros. Descritor de arquivo (File Descriptor) 2 (Ex. cat 98032983927 2> /dev/null)

Mais informações sobre Entrada/Saída e Streams



#### File Descriptor

Abstração de uma identificação para acessar um arquivo. Valor inteiro, cada processo possui um *file descriptor table*.

Pertence ao processo

file descriptor

global file table

read-only, offset: 0

write-only, offset: 0

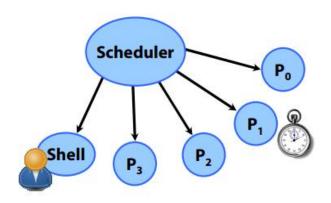
yath/myfile.txt

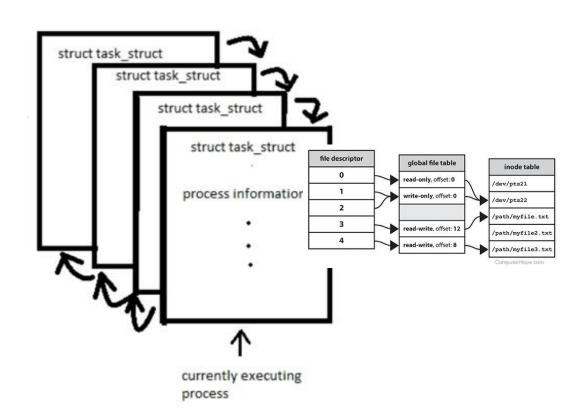
/path/myfile2.txt

/path/myfile3.txt

Mais informações sobre *file descriptor* 

### Pausa um entendimento geral





#### Tudo são arquivos

Vamos para o terminal.

- 1. -: regular file
- 2. d: directory
- 3. c: character device file
- 4. b : block device file
- 5. s:local socket file
- 6. p:named pipe
- 7. l : symbolic link





#### File Descriptor

Abstração de uma identificação para acessar um arquivo. Valor inteiro, cada processo possui um *file descriptor table*.

Pertence ao processo

file descriptor

global file table

read-only, offset: 0

write-only, offset: 0

yath/myfile.txt

/path/myfile2.txt

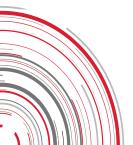
/path/myfile3.txt

Mais informações sobre *file descriptor* 

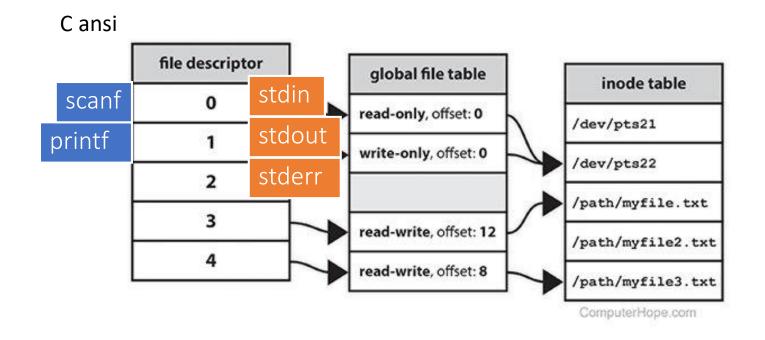
## Redirecionamento via código

Vamos para o Visual Studio Code!

Vamos implementar um código de leitura e escrita de arquivos.



#### Redirecionamento via código





### dup e dup2

dup: cria uma cópia de um descritor de arquivo existente e aloca no menor número de descritor disponível no processo

dup2: cria uma cópia de um descritor de arquivo existente e aloca no valor informado como parâmetro na função



#### Desafio da Semana

Desenvolver uma aplicação em C que possa ser utilizada no terminal para modificar caracteres de um texto para caixa alta.

Ex. lshw | sua\_aplicação

Espera-se que todo o texto de saída esteja em caixa alta.



#### Consolidando

- Compreender o que é um *File Descriptor*
- Utilizar redirecionamento via terminal e via código
- Utilizar dup e dup2
- Compreender o que são sistemas POSIX





FIM

Tiago Sanches da Silva