

# Programação em *shell*

## Processos, *foreground* e *background*

André Grégio, Fabiano Silva, Luiz Albini e Marcos Castilho

Departamento de Informática – UFPR, Curitiba/PR

21 de julho de 2020

# Objetivos

- Apresentar o conceito de processos
- Mostrar que eles podem rodar em *foreground* ou em *background*
- Mostrar as principais teclas de controle de processos

## *Background e foreground*

- O UNIX é um sistema multitarefas, vários *jobs* podem rodar ao mesmo tempo
- Existem processos simultâneos de muitos usuários, inclusive vários deles são do seu usuário
- Os processos podem rodar mesmo você não estando logado

## *Background e foreground*

- Quando você digita um comando, seu *prompt* fica preso até que o comando termine
- Este *job* está rodando em *foreground*
- Você pode colocá-lo em *background* colocando um & após o comando
- Isto vai liberar seu *prompt* para você digitar outros comandos enquanto o primeiro roda.

## Exemplo

Neste exemplo o libreoffice foi chamado em *background*, o *prompt* é liberado. A *shell* mostra o número do processo que está rodando, no caso 4252. O comando *jobs* mostra todos os seus processos que estão em *background*. Quando o processo termina a *shell* avisa.

```

ci1001@fradim:~/tmp$ libreoffice &
[1] 4252
ci1001@fradim:~/tmp$ <prompt liberado>
ci1001@fradim:~/tmp$ jobs
[1]+  Executando                libreoffice &
ci1001@fradim:~/tmp$
[1]+  Concluído                  libreoffice
ci1001@fradim:~/tmp$
  
```

## Entrada e saída em *background*

- Não se deve ter E/S com *jobs* em *background*
- Se tiver entrada, o *job* vai ficar esperando
- Se tiver saída, sua *shell* ficará poluída com a saída e vai te atrapalhar
- Por isso é aconselhável usar redirecionamento de E/S nestes casos

```
$ comando < entrada_comando > saida_comando
```

# Prioridade de processos

- Todo processo no Linux tem uma prioridade definida pelo Sistema Operacional.
- O usuário pode alterar esta prioridade com o comando *nice*, como forma de boa educação quando tem um processo demorado para não atrapalhar outros usuários.
- Os comandos *top* ou *ps* permitem ver a prioridade atual dos processos

# Teclas de controle

- Algumas teclas de controle podem ser usadas para ajudar a controlar processos
- A lista padrão está no próximo slide
- Pode-se modificar estas teclas

## Lista de teclas de controle

CTRL-C	interrompe o comando atual
CTRL-D	fim da entrada
CTRL-\	interrompe o comando atual
	se CTRL-C não funcionar
CTRL-S	interrompe saída na tela
CTRL-Q	restaura saída na tela
DEL ou CTRL-?	apaga caracter
CTRL-U	apaga toda a linha de comando
CTRL-Z	suspende o processo
CTRL-L	limpa a tela

# Conclusão

- Apresentamos os conceitos processos
- Mostramos que eles podem rodar em *foreground* ou em *background* e como trocar de um modo para o outro
- Também mostramos as teclas de controle para processos
- Na próxima videoaula veremos os símbolos especiais da *shell*

# Licença

- Slides feitos em  $\text{\LaTeX}$  usando beamer
- Licença

*Creative Commons* Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>

Creative Commons Atribuição-Uso Não-Comercial-Vedada a Criação de Obras Derivadas 2.5 Brasil License.<http://creativecommons.org/licenses/by-nc-nd/2.5/br/>