

Lista de Exercícios 06

Expansões

Exercício 01) Considere que só existem os seguintes arquivos no diretório de trabalho atual.

```
$ ls
intro  notesb  ref2  section1  section3  section4b
notesa  ref1    ref3  section2  section4a  sentrev
```

Dê comandos para cada uma das seguintes situações, usando caracteres coringa para expressar nomes de arquivos **usando uma expansão (sem espaços) com o menor número de caracteres possíveis**. Eles devem funcionar especificamente para os arquivos listados acima e não genericamente para quaisquer arquivos.

- a) Listar todos os arquivos que começam com section.

```
ls *c*
```

- b) Listar somente os arquivos section1, section2 e section3.

```
ls *c*[1-3]
```

- c) Listar apenas o arquivo intro.

```
ls i*
```

- d) Listar os arquivos section1, section3, ref1 e ref3.

```
ls *[13]
```

Exercício 02) Considere que os seguintes arquivos estão no diretório de trabalho atual.

```
$ ls
abc abd abe abf abg abh
```

- a) Explique porque acontece a seguinte mensagem de erro.

```
$ rm abc ab*
rm: cannot remove 'abc': No such file or directory
```

Apresenta erro devido a expansão ter sido feita antes da remoção do arquivo abc, então ele tenta remover o arquivo abc duas vezes.

- b) Quais arquivos seriam exibidos se fosse executado o comando `ls` subsequentemente?

A pasta estará vazia pois todos os elementos foram removidos porem ele tentou remover um duas vezes.

Exercício 03) A expansão de nomes de arquivos respeita o comportamento de não exibir arquivos escondidos (aqueles que começam com ponto) e não os expande. Para exibi-los é preciso forçar o casamento do primeiro caractere ponto com o restante do nome de arquivo.

```
$ ls -d .*  
.  
..  
.CFUser  
.DS_Store  
.Trash  
.argouml  
.bash_history  
.cache  
.config  
.cups  
.dropbox  
.lessht  
.profile  
.swipl-dir-history
```

Contudo, tal expansão também exibiu o diretório atual (.) e o anterior (..). Construa **uma expansão (sem espaços)** de forma a mostrar todos os arquivos escondidos, mas excluindo os diretórios atual e anterior (sem *pipe*).

```
$ ls -d .{[!]*,.*}  
.CFUser  
.DS_Store  
.Trash  
.argouml  
.bash_history  
.cache  
.config  
.cups  
.dropbox  
.lessht  
.profile  
.swipl-dir-history
```

Exercício 04) Considere o diretório de trabalho atual que possui arquivos de vídeo com extensões avi, mp4 e rmvb, e de legenda com extensão srt de suas séries favoritas. Contudo, os arquivos de legenda não estão com a extensão padronizada, usando diversas variações de caixa alta e baixa.

```
$ ls  
battlestar.galactica.s01e01.avi  
battlestar.galactica.s01e01.sRt  
breaking.bad.s01e04.Srt  
breaking.bad.s01e04.mp4  
game.of.thrones.s01e02.SRT  
game.of.thrones.s01e02.mp4  
house.of.cards.s01e05.rmvb  
house.of.cards.s01e05.sRT  
lost.s01e06.SrT  
lost.s01e06.avi  
x-files.s01e03.avi  
x-files.s01e03.srt
```

Complete o comando a seguir com **uma expansão (sem espaços)** para exibir somente os arquivos de legendas, mostrando todos eles. Deve funcionar para qualquer situação, não somente para o exemplo dado acima.

```
$ ls *[Ss][Rr][Tt]
```

Exercício 05) A série *24 horas* foi feita originalmente com 8 temporadas, em que cada temporada possui exatamente 24 episódios, um para cada hora do dia do personagem principal. Complete o comando a seguir com **uma expansão (sem espaços)** para criar a estrutura de diretórios para guardar cada episódio seguindo a padronização 24/SeasonNN/XX, tal que NN é o número da temporada com dois dígitos e XX do episódio com dois dígitos.

```
$ mkdir -p 24/Season0{1..8}/{0{1..9},1{0..9},2{0..4}}
```

Exercício 06) O comando a seguir foi utilizado para passar o diretório atual

(pwd) como parâmetro para o comando basename.

```
$ pwd | xargs basename
```

Reescreva esse comando para passar o parâmetro **usando substituição de comandos**, ao invés da combinação *pipe* com xargs.

```
$ basename $(pwd)
```