

Comandos Avançados

João Marcelo Uchôa de Alencar

Universidade Federal do Ceará - Quixadá

29 de Agosto de 2019

ed - Editor de texto orientado a linhas

- ▶ Uso minimalista.
- ▶ Utiliza as mesmas expressões regulares que outros comandos e linguagens.
- ▶ Ele é usado para exibir, modificar e manipular arquivos de texto.

Indicador de Pesquisa

Para abrir um arquivo:

```
$ ed [nomedoarquivo]
```

A única coisa que será impressa é a quantidade de *bytes*.

Como é que uso isso ?!?

Digite /, e em seguida uma cadeia de caracteres.

Para imprimir todo o texto: 1,\$ p

Procurar início da linha: ^cadeia

Procurar fim da linha: cadeia\$

Para sair do editor: q

Prompt de Comando

Para tornar as coisas mais fáceis, inicie o ed com o prompt de comando:

```
$ ed -p:
```

Podemos criar um alias para facilitar: **\$ alias ed='ed -p:'**

Para adicionar texto a um *buffer*: **a**

Para voltar ao *prompt* de comando, deixe linha com apenas um '.'.

Para salvar em um arquivo: **w nomedoarquivo**

Imprimir as linhas numeradas: **1,\$ n**

Ir para uma determinada linha: **n**, onde n é o número da linha.

Alterar o conteúdo da linha: **c**

Substituição

Já deve ter notado que **1,\$** representa uma faixa de valores.

Nesse caso, é da linha um até o final do arquivo.

Porém poderia ser qualquer faixa: **1,5 1,10 5,15** etc

O **p** é função imprimir, o **n** numerar as linhas, etc

O **g** permite a busca não por cadeias, mas sim expressões regulares: **g/expressao /p**

O **s** é a função substituir: **1,\$ s/expressao /cadeia /g**

Observer o **g** no comando anterior.

O comando sed

```
sed expressão regular [arquivo]
```

A expressão regular é do formato:

```
[<endereço-1>, [<endereço-2>]] <função> [argumento]
```

Funções

Substituir por cadeia:

```
sed 's/expressaoregular/cadeia/' arquivo
```

Substituir por caractere:

```
sed 'y/caractere-1/caractere-2/' arquivo
```

Imprimir linhas que casam com expressão:

```
sed '/expressao/p' arquivo
```

Deletar linhas:

```
sed '1,4d' arquivo  
sed '/expressao/d' arquivo
```

Adicionar uma linha em posição:

```
sed '1a \novalinha' arquivo
```

A opção -n reduz o que é impresso na tela, enquanto a opção -i atualiza o arquivo.

A opção -r ativa suporte a metacaracteres avançados nas expressões regulares.

cut - Cortando cadeias de caracteres

```
cut -f[campo] -d[delimitador] arquivo
```

O *delimitador* indica qual caractere é usado para separar os campos.

O número campo indica qual campo deve ter a coluna impressa.

tr - Transformar cadeias de caracteres

```
tr [opções] <caracteresoriginais> <caracteresdestino>
```

Converte dos caracteres originais para os caracteres de destino de acordo com a ordem que aparecem.

Pode usar as classes POSIX.

A opção -s elimina caracteres repetidos.

uniq - Eliminando repetições

`uniq` arquivo

Deixa uma única cópia para cada linha repetida.
As linhas devem ser adjacentes.

Aritmética no Bash

```
expr <operando1> <operador> <operando2>
```

O operador pode ser +, -, *, /, %

É extremamente sensível aos espaços.

Aritmética no Bash

`bc [arquivo]`

É como se fosse um *bash* para aritmética.

Mais utilizado como comando destino de um pipe.

Aceita o parâmetro *scale*.

Aritmética

Também é possível usar o próprio Bash para expressões aritméticas inteiras.

```
$(( expressão ))
```

No lugar de executar um comando, o parênteses extra calcular a expressão.

Exercícios - Parte I

Continuando no diretório *atividades/atividade04*. Copie o arquivo */home/compartilhado/atividade04.py* para o diretório criado. Crie o arquivo *parte_01.txt* e coloque os comandos *sed* que realizam as seguintes ações:

1. Troque o interpretador *Python* para */usr/bin/python3*.
2. Em um único comando, altere o nome das variáveis *nota1*, *nota2* e *notaFinal* para usarem apenas letras maiúsculas (por exemplo, *nota1* vira *NOTA1*).
3. Altere o arquivo para importar o módulo *time* do *Python*.
4. Adicione um comando ao final do programa para imprimir a data atual (*time.ctime()*).

Exercícios - Parte II

Ainda no diretório *atividades/atividade04*, crie o arquivo *parte_02.txt*. Usando os comandos aprendidos, crie uma sequência de comandos ligados por `|` que realize as ações abaixo. Coloque uma sequência por linha no arquivo *parte_02.txt*.

1. Analisando `/etc/passwd`, liste, um por linha, os diferentes shells utilizados pelos usuários (último campo separado por `:`).
2. Veja o arquivo `/home/compartilhado/despesas.txt`. Use os comandos para calcular o total de despesas.