

Capítulo 6

===Exercício 1===

Eu preciso inserir textos (diversos) em um espaço com largura restrita e para conseguir isso tenho que desenvolver um *script* para dividir os textos de forma a que eles caibam no espaço que tenho.

Essa rotina receberá os seguintes parâmetros:

- **\$1** = Tamanho máximo no qual o texto será partido;
- **\$*** = O texto que será partido;
- Observações:
 - O texto só será partido nos espaços em branco;
 - Por ser feita para ser usada pelo programador, o *script* não critica se existe espaço antes do tamanho definido.

Observações:

1. O texto sempre será dividido nos maiores pedaços que couberem no tamanho indicado por **\$1**;
2. Nos locais de quebra de linha, o *script* deverá trocar o espaço em branco por um **\n**.

```
$ cat exerc6-1.sh
```

```
#!/bin/bash
# Exercício 1 do capítulo 6
# Parte um texto nos espaços para caberem numa largura
#+ Recebe os seguintes parâmetros:
#   +- $1 = Tamanho máximo no qual o texto será partido
#   +- $* = O texto que será partido
#   +- Obs: O texto só será partido nos espaços em branco
#   +- Por ser feita para programador, a função não
#   +- critica se existe espaço antes do tamanho definido
Larg=$1
shift      # Jogou fora a largura. Agora ficou só o texto
LinParc=   # Linha parcial aquela que terá o tamanho < que $1
LinTot=    # Linha completa à qual agregaremos cada LinParc
for Palavra in $@
do
    (( ${#LinParc} + ${#Palavra} > $Larg )) && {
        LinTot+=" $LinParc\n"
        LinParc=
    }
    LinParc+=" $Palavra "
done

LinTot+=" $LinParc"
echo -e "$LinTot"
```

===Exercício 2===

Fazer um programa que leia da tela os seguintes dados:

- Nome da Máquina.
- *Login Name* do operador naquela Máquina.
- Senha do operador.
- Nome dos arquivos que serão transmitidos.

Observações:

A senha do operador não poderá ser ecoada na tela.

Será permitido o uso de metacaracteres na constituição do nome dos arquivos.

A leitura de nomes de arquivos será feita em *loop*, até que seja teclado um nome vazio.

Os nomes dos arquivos informados deverão formar uma lista que será exibida no centro da antepenúltima linha do terminal.

Coringas podem ser informados para definir nomes e arquivos, por exemplo: **arq***

```
$ cat exerc6-2.sh
```

```
#!/bin/bash
# Exercício 2 do capítulo 6
#+ Loop de leitura de dados e manipulação de cadeias
# Para efeito didático, esse exercício será
#+ feito usando uma tela pré formatada.

# Montagem da tela
clear
echo "
                Nome da Máquina:

                Login Name na máquina:

                Senha:

                Nome dos arquivos:"

# Para formatar a tela usa-se somente um echo, não sendo
#+ necessário usar um echo para cada linha, mas note que abrimos
#+ aspas está após o echo e o fechamos no fim da última linha
tput cup 1 36
read Maq
tput cup 3 36
read LName
tput cup 5 36
read -s Sen
# Loop de leitura dos nomes de arquivos
```

```

Lista=
LinhaMsg=$((tput lines) - 3)) # 3 linhas acima da última
while true
do
    tput cup 7 36
    tput el # Apaga o nome do arquivo anterior
    read Arq
    [[ -z $Arq ]] && break # Sai do loop se nada foi informado

    # Normalmente o teste a seguir seria: [[ -e $Arq ]],
    #+ porém se $Arq possuir um coringa, sua expansão produzirá
    #+ um erro, pq a opção -e é unária (só aceita um argumento)
    ls $Arq > /dev/null 2>&1 || {
        tput cup 7 51
        echo "$Arq não existe. Tecle algo..." >&2
        read -n1
        continue # Volta para ler outro arquivo
    }
    Lista+=$(echo $Arq)' ' # echo para expandir coringas
    ColMsg=$((tput cols)-${#Lista})/2 # Centrando a lista
    tput cup $LinhaMsg $ColMsg
    echo $Lista
    # Não estamos controlando tamanho da lista, pode estourar a linha
done

```

===Exercício 3===

Vamos executar o comando **seq -w 100** e para listá-los vamos ver a largura da sua tela, subtrair 10 (para fazer as margens) e dividi-la por 4 e então proceder de duas formas diferentes:

- A) Listar o resultado em 5 colunas separadas pelo resultado desta divisão, começando na coluna 4 da tela (margem esquerda). Dessa forma a primeira coluna teria de **001** a **020** e a última de **081** a **100**;
- B) Novamente usando a largura obtida na divisão listar a partir da coluna 4 da tela de **001** a **005** na primeira linha e assim sucessivamente até termos **096** a **100** na última linha.

Dê um **read** para parar entre as duas listagens, mas se você quiser, faça seu *script* aceitando as opções **-v** (vertical), como definido no item A ou **-h** (horizontal), como no item B.

```
$ cat exerc6-3.sh
```

```

#!/bin/bash
# Exercício 3 do capítulo 6
# Redireciona, de acordo com a opção para:
#+ Opção   Programa   Tipo Listagem
#+ -v      exerc6.3.1  Vertical
#+ -h      exerc6.3.2  Horizontal
#+ Sem     Ambos      Ambos

# Cálculo dos 4 intervalos, entre as 5 colunas,
#+ Total de colunas - 25 (10 das margens + 5 colunas de 3 char)

```

```

export Intervalo=$((($(tput cols) - 25) / 4)) # Para os filhos verem
(($# == 0)) && {
    exerc6.3.1
    read -n1 -p ' Tecle algo... '
    Col=4
    exerc6.3.2
    exit
}
[[ $1 == -v ]] && {
    exerc6.3.1
    exit
}
[[ $1 == -h ]] && {
    exerc6.3.2
    exit
}
}
echo Parâmetro Inválido >&2

```

\$ cat exerc6-3-1.sh

```

#!/bin/bash
# Exercício 3 do capítulo 6
# Faz a listagem vertical baseada no intervalo
#+ entre colunas exportado pelo exerc6.3, seu pai
clear
Col=4
seq -w 100 | xargs -n20 | # 100/5=20, isto é, 20 linhas ao todo
while IFS=$'\n' read Vinte
do
    Lin=1
    for Num in $Vinte
    {
        tput cup $Lin $Col
        echo $Num
        let Lin++
    }
    let Col+=Intervalo+3
    #sleep 0.3 # Para câmara lenta, descomente o inicio da linha
done

```

\$ cat exerc6-3-2.sh

```

#!/bin/bash
# Exercício 3 do capítulo 6
# Faz a listagem horizontal baseada no intervalo
#+ entre colunas exportado pelo exerc6.3, seu pai
Col1=4
Col2=$((Col1+Intervalo+3)) # 3 = largura de cada número listado
Col3=$((Col2+Intervalo+3))
Col4=$((Col3+Intervalo+3))
Col5=$((Col4+Intervalo+3))
clear
Lin=1
# Fazendo a listagem horizontal
seq -w 100 | xargs -n5 |

```

```
while IFS=' ' read n1 n2 n3 n4 n5
do
    tput cup $Lin $Col1; echo $n1
    tput cup $Lin $Col2; echo $n2
    tput cup $Lin $Col3; echo $n3
    tput cup $Lin $Col4; echo $n4
    tput cup $Lin $Col5; echo $n5
    let Lin++
done
```

===Exercício 4===

Ler cada um dos números gerados pelo comando `seq -w 200` listando um por linha nas colunas que ficam no centro da tela.

A cada `<quantidade de linhas da tela> - 3` parar dando a mensagem **"Tecle algo para prosseguir ou f para fim..."** e esperando teclar somente 1 caractere. Se for teclado `f` ou `F`, sair do programa. Qualquer outro caractere, listar os próximos `<quantidade de linhas da tela> - 3`

Imagine o seguinte exemplo:

```
$ tput cols
113
$ tput lines
28
```

Nesse caso, seu *script* escreveria a saída do `seq -w 200` na coluna 55, calculada subtraindo-se a largura da coluna dos dados (3) da largura da tela (113) e dividindo esse resultado por 2. Toda vez que chegasse na linha 25, daria a mensagem **"Tecle algo para prosseguir ou f para fim..."**

```
$ cat exerc6-4.sh
#!/bin/bash
# Exercício 4 do capítulo 6
# Gera uma lista no centro da tela e parando
#+ 3 linhas antes do final da tela esperando
#+ usuário pedir para encerrar ou não
Centro=$((($tput cols)-3)/2) # Calculando centro da tela
UltLin=$((($tput lines)-3)) # Calculando última linha para
listagem
LinAtu=0 # Contador da linha atual
clear
seq -w 200 |
while read Num
do
    tput cup $((++LinAtu)) $Centro # Posicionou para listar
    echo $Num
    ((LinAtu == UltLin)) && { # Chegamos na última linha?
        read -n1 -p "Tecle algo para prosseguir ou f para fim..."
    }
    < /dev/tty
    [[ $REPLY == F || $REPLY == f ]] && {
        echo # Para o prompt ser na linha seguinte
    }
done
```

```
        exit
    }
    LinAtu=0          # Zera o contador e limpa tela
    clear
}
done
```