

Capítulo 5

===Exercício 1===

Imagine uma empresa com uma filial em cada estado espalhados pelo Brasil e um arquivo chamado **ArqOLs** que tem cadastrados os operadores de todos os escritórios estaduais, responsáveis pela leitura de todos os *e-mails* recebidos e pela tomada da atitude solicitada por cada *e-mail*. Este arquivo tem o *layout* a seguir:

```
<Nº Filial><TAB><Nome da Máquina><TAB><Oper1> <Oper2> ... <Opern>
```

Faça um programa que mande um *e-mail* para todos os operadores de uma determinada filial (recebendo **<Nº OL>** ou **<Nome da Máquina da filial>** como parâmetro) com o conteúdo de um arquivo (nome recebido como segundo parâmetro).

Obs.: Não se esqueça que dentro da mesma rede, o comando para passar *e-mail* tem o seguinte formato:

```
mail operador@máquina < arquivo
```

```
$ cat exerc5-1.sh
```

```
# Capítulo 5 exercício 1
# Pesquisa um arquivo por parâmetro passado
#+ e depois faz um loop para envio de e-mail

# Primeiramente vejamos se o parâmetro passado é
#+ numérico, ou seja OL, ou não, quando seria máquina
if expr $1 + 0 >/dev/null 2>&1
then
    # if bem sucedido então é numérico
    Reg=$(grep "^$1 " ArqOLs) || { # Após $1 tem <TAB>
        echo $0: Informe nº OL ou nome MÁQUINA >&2
        exit 1
    }
    echo Recebi parâmetro numérico # Esta linha pode ser apagada
else
    # De erro no expr então $1 é alfa, ou seja máquina
    Reg=$(grep " $1 " ArqOLs) || { # $1 entre <TABs>
        echo $0: Informe nº OL ou nome MÁQUINA >&2
        exit 2
    }
    echo Recebi parâmetro alfabético # Esta linha pode ser apagada
fi

# Se o programa não abortou, o registro está em $Reg
#+ então vamos trabalhá-lo:
Maquina=$(cut -f2 <<< "$Reg") # As aspas são para que os <TAB>
Operadores=$(cut -f3 <<< "$Reg") #+ não se transformem em espaços
# Como os operadores são uma lista separada
#+ por espaços, o indicado é um loop de for:
for Operador in $Operadores
do
    echo Enviando e-mail para $Operador
    # mail $Operador@$Maquina < ARQUIVO
```

```
# Se vc estivesse nessa rede, bastaria descomentar
#+ a linha do mail e escolher o nome do ARQUIVO
done
```

===Exercício 2===

Suponha um cartão de crédito com o número 1543 2109 8765 4321

Para verificar se esse número está correto devemos usar o algoritmo de *Luhn*, que é calculado da seguinte forma:

Ordem	↓ ↓ ↓ ↓ ↓ ↓								
	1 1 1 1 1 1								
	5 4 3 2	1 0 9 8	7 6 5 4	3 2 1 0					
Número do Cartão de Crédito	1 5 4 3	2 1 0 9	8 7 6 5	4 3 2 1					
Mantém	5 3	1 9	7 5	3 1					
Dobro	2 8	4 0	1 1	6 2	8 4				
Valor para soma	2 5 8 3	4 1 0 9	7 7 3 5	8 3 4 1					

Soma=70

Ou seja, os valores de ordem ímpar (ordem zero (0) é o mais à direita) são multiplicados por 2 e no caso um destes produtos ser maior que 9, deve ser subtraído de 9. Após isso devemos fazer a soma de cada um desses valores calculados, com cada algarismo de ordem par.

O número do cartão de crédito estará correto caso a soma obtida seja múltipla de dez.

Experimente agora usar o número do seu cartão de crédito (mas por motivo de segurança, ao final dê **unset** na variável que você usou para armazenar o número ou encerre a sua seção de *Shell*).

Se você quiser ir um pouco mais fundo, também te adianto o seguinte acerca dos 2 primeiros algarismos do número:

2 primeiros algarismos Operadora

4?

Visa

3[47]

American Express

5[15]

MasterCard

Obviamente existem muito mais operadoras que essas 3, inclusive para cartões de lojas, de milhagem, ... Mas conseguindo fazer para esses três, que são os mais importantes, é só pesquisar e incrementar os que interessarem.

```
$ cat exerc5-2.sh
```

```
#!/bin/bash
```

```
# Capítulo 5 exercício 2
```

```
# Calcula DV e indica operadora de cartão de débito/crédito
```

```

(( $# != 1 )) && {
    echo 'Uso: $0 <NUMERO DO CARTAO SEM O ÚLTIMO ALGARISMO>' >&2
    exit 1
}

[[ ${#1} =~ 1[45] ]] || { # Só aceita cartão com 14 ou 15 algarismos
    echo 'Uso: $0 <NÚMERO CARTÃO SEM DV>' >&2
    exit 1
}

case ${1:0:2} in
    # 2 primeiros algarismos para descobrir operadora
    4?) Oper=Visa ;;
    3[47]) Oper=American\ Express ;;
    5[15]) Oper=MasterCard ;;
esac

printf -v Num '%015i' $1 # Inclui 0 esquerda para American Express
Soma=0
Mul=212121212121212
for ((Pos=0; Pos<${#Num}; Pos++))
{
    let Alg=${Num:$Pos:1}*${Mul:$Pos:1} # Pode-se fazer o mesmo usando cut
    Alg=$((Alg>9 ? Alg-9 : Alg)) # Se Numero > 9 soma seus algarismos
    let Soma+=Alg
}

echo "Operadora do cartão é $Oper
E o DV é $(((10-Soma%10)%10))"

```