

INSTITUTO POLITÉCNICO DE TOMAR

Escola Superior de Tecnologia

Departamento de Engenharia Informática



SISTEMAS OPERATIVOS

Trabalho Prático

Scripts CShell

2003/2004

José Nicolau (5205)
Tiago Pina (5223)

```
#!/bin/csh
#-----
# Autor      : Jose Nicolau (5205) & Tiago Pina (5223)
#            IPT - EI/SO 2003/2004
#
# Uso        : trbl
#
# Ficheiros
#   Usados  : funcionarios.dat, medicos.dat, relaciona.dat
#
# Descricao  : Gestão de uma pequena base de dados em ficheiros ASCII - implementa
#            algumas operações básicas sobre os ficheiros de dados, como a
#            introdução e remoção de dados, listagem e pesquisa
#-----

## inicializacao da base de dados caso não exista
if (! -e funcionarios.dat) then
  echo "0op;SAM;SAM" > funcionarios.dat
endif
if (! -e medicos.dat) then
  echo "0md;SAM;SAM;SAM" > medicos.dat
endif
if (! -e relaciona.dat) then
  echo "" > relaciona.dat
endif

## ----- menu - principal
set opt = 0

while ($opt != 9)
  clear
  echo "SAM DataBase - Menu de Operacoes "
  echo " "
  echo "1 - inserir novo funcionario"
  echo "2 - inserir novo medico"
  echo "3 - registar uma consulta (funcionario/medico)"
  echo "4 - listar todos os funcionarios"
  echo "5 - listar todos os medicos"
  echo "6 - pesquisar"
  echo "7 - remover ficha do funcionario e respectivas consultas"
  echo "8 - remover ficha do medico e respectivas consultas"
  echo " "
  echo "9 - sair"
  set opt = $<
  clear
  switch ($opt)
  # -----
  case "1" :
    echo "1 - Inserir novo Funcionario"
    echo " "
    echo "Introduza o nome do funcionario:>"
    set nomef = $<
    set nomef = `echo "$nomef" | tr '[a-z]' '[A-Z]` # Transforma o nome em Maiusculas
    set ef = `grep "$nomef" funcionarios.dat | wc -l` #Existe funcionario
    if ( $ef != 0 ) then # nao ; processa os outros dados
      echo "--> FUNCIONARIO JA EXISTE <--"
    else
      echo "Introduza a morada:>"
      set moradaf = $<
      if (-e funcionarios.dat) then
        # obter o ultimo elemento do ficheiro
        set idf = `tail -1 funcionarios.dat | awk -F"op" '{print $1}` # obter o nº
        @ idf = $idf + 1 # + 1 caso exista
      else
        set idf = 1 # 1 se não existir
      endif
    endif
  endswitch
endwhile
```

```

# Fazendo a formatacao do id para 000op, estamos a limitar o n. de
# funcionarios, por isso nao a usamos.
# para a implementacao desta estrutura de bastaria usar
#
# if (idf<10)
# set idf = `echo "00$idf"`
# else
#   if (idf < 100) then
#     set idf = `echo "0$idf"`
#   endif
# endif
#
# este codigo seria tb usado na numeracao dos medicos (próximo case)
set reg = `echo "$nomef;$moradaf" | tr '[a-z]' '[A-Z]'\` # criar registo
echo "$idf"op;$reg" >> funcionarios.dat #completar registo e gravar no ficheiro
endif
breaksw

# -----
case "2" : #funcionamento semelhante ao da inserção de funcionarios
echo "2 - Inserir nome Medico"
echo " "
echo "Introduza o nome do medico :>"
set nomem = $<
set nomem = `echo "$nomem" | tr '[a-z]' '[A-Z]'\` # obter e verificar se existe o
set em = `grep "$nomem" medicos.dat | wc -l` # funcionario
if ($em != 0) then
echo "--> MEDICO JA EXISTE <--"
else
echo "Introduza a especialidade :>"
set espm = $<
echo "Introduza a clinica :>"
set clinica = $<
if (-e medicos.dat) then
set idm = `tail -1 medicos.dat | awk -F"md" '{print $1}' `
@ idm = $idm + 1
else
set idm = 1
endif
set reg = `echo "$nomem;$espm;$clinica" | tr '[a-z]' '[A-Z]'\` #construir e gravar
echo "$idm"md;$reg" >> medicos.dat # o registo
endif
breaksw

# -----
case "3":
echo "3 - Registrar uma consulta"
echo "Introduza o nome do funcionario :>"
set nfunc = $<
set nfunc = `echo "$nfunc" | tr '[a-z]' '[A-Z]'\`
set ef = `grep "$nfunc" funcionarios.dat | wc -l`
if ($ef == 0) then
echo "FUNCIONARIO NAO EXISTE !!!"
else
set numf = `grep "$nfunc" funcionarios.dat | awk -F";" '{print $1}'`
echo "Necessita de uma lista dos codigos dos medicos e sua especialidade ? (S/N)"
set opt1 = $<
if ($opt1 == 'S') then # mostra a lista de medicos
cat medicos.dat | tr ';' '\t\t' # substituindo ; por \t\t (2 tab)
endif
echo "Introduza o codigo do medico :>"
set cmed = $<
echo "Introduza a data da consulta :>"
set dcon = $<
echo "Custo da consulta :>"
set cust = $<
echo "$numf;$cmed;$dcon;$cust" >> relaciona.dat #grava o registo
echo "Consulta introduzida ..."
endif
breaksw

```

```
# -----
case "4":
    echo "Id;Nome;Morada" | tr ';' '\t\t' #mostra o cabeçalho da tabela
    cat funcionarios.dat | tr ';' '\t\t' | more #lista os dados
    breaksw
# -----
case "5":
    echo "Id;Nome;Especialidade;Clinica" | tr ';' '\t\t' #mostra o cabeçalho
    cat medicos.dat | tr ';' '\t\t' | more # da tabela e lista os dados
    breaksw
# -----
case "6" : #SUBMENU Pesquisa
    echo "6 PESQUISA"
    echo " "
    echo " 1 - Consultas por funcionario e suas despesas"
    echo " 2 - Consultas por medico e sua facturacao"
    echo " 3 - Despesas numa dada especialidade"
    echo " 4 - Consultas efectuadas a partir de uma dada data"
    echo " "
    set opt1 = $<
    clear
    switch ($opt1)
# -----
    case '1' :
        echo "61 Consultas do funcionario"
        echo " "
        echo "Introduza o nome do funcionario :>"
        set nomef = $<
        set nomef = `echo "$nomef" | tr "[a-z]" "[A-Z]"` # obter nome em maius.
        echo "Consultas"
        echo " "
        set ef = `grep "$nomef" funcionarios.dat | wc -l` # verificar se existe
        if ($ef>0) then # sim ?
            set idf = `grep "$nomef" funcionarios.dat | cut -f1 -d';'` # obter id
            set cons = `grep "$idf" relaciona.dat` # obter todas as consultas
            set total = 0
            foreach x ($cons) # para cada consulta
                set idm = `echo "$x" | cut -f2 -d';'` # obter id do medico
                set esp = `grep "$idm" medicos.dat | cut -f3 -d';'` #obter especial.
                echo $x | sed -e "s/$idm/$esp/" | awk -F";" '{printf $3" - "$2" -
"$4"\n"}' # listar data - especialidade - valor
                @ total = $total + `echo $x | awk -F";" '{printf $4}'` # calc. total
            end
            echo " "
            echo "TOTAL de Despesas = $total" # mostra total
        endif
        breaksw
```

```
# -----
case '2' :
    echo "62 Consultas dos Medicos"
    echo " "
    echo "Introduza o nome do medico :>"
    set nomem = $<
    set nomem = `echo "$nomem" | tr "[a-z]" "[A-Z]"`
    echo "Consultas"
    echo " "
    set em = `grep "$nomem" medicos.dat | wc -l` # verifica se existe
    if ($em>0) then # sim ?
        grep "$nomem" medicos.dat | cut -f3 -d';' # mostra especialidade
        set idm = `grep "$nomem" medicos.dat | cut -f1 -d';'` # obter id medico
        set cons = `grep "$idm" relaciona.dat` #obter todas as consultas do med.
        set total = 0
        echo " "
        foreach x ($cons) # para cada consulta
            set idf=`echo "$x" | cut -f1 -d';'` # obter id func.
            set nomef=`grep "$idf" funcionarios.dat | cut -f2 -d';'` #obter nome
            echo $x | sed -e "s/$idf/$nomef/" | awk -F";" '{printf $3" - "$1" -
"$4"\n"}' # listar data - doente - valor
            @ total = $total + `echo $x | awk -F";" '{printf $4}'` # calc. total
        end
        echo " "
        echo " Ganhos do Medico = $total" #apresenta o total
    endif
breaksw

# -----
case '3' :
    echo "63 Despesas por Especialidade"
    echo " "
    echo "Introduza a especialidade :>"
    set esp = $<
    set esp = `echo $esp | tr "[a-z]" "[A-Z]"`
    set total = 0
    set medicos = `grep $esp medicos.dat` # obter todos os medicos com a esp.
    foreach x ($medicos)
        set idm = `echo $x | cut -f1 -d';'` # obter id do medico
        set cons = `grep "$idm" relaciona.dat` # obter todas as consultas
        foreach y ($cons) # para cada consulta
            echo $y | awk -F";" '{printf $3" - "$4"\n"}' # mostrar data - valor
            @ total = $total + `echo $y | awk -F";" '{printf $4}'` # calc total
        end
    end
    echo " "
    echo " Total de despesas = $total" # mostra total
breaksw
```

```

# -----
case '4' :
    echo "64 Consultas a partir de dada Data "
    echo " "
    echo "Introduza a data (ddmmyyyy) :>"
    set data = $<
    set c = `cat relaciona.dat` #obter todas as consultas
    set cons = " "
    foreach k ($c)
        set d = `echo "$k" | cut -f3 -d';'` # obter data da consulta
        if ($d >= $data) then # se >= então guarda
            set cons = `echo "$cons $k"`
        endif
    end
    set total = 0
    foreach x ($cons) # para consulta armazenada
        set idf = `echo $x | cut -f1 -d';'` # obtem o id de funcionario
        set idm = `echo $x | cut -f2 -d';'` # obtem o id de medico
        set nm = `grep $idm medicos.dat | cut -f2 -d';'` # obter o nome do med.
        set esp = `grep $idm medicos.dat | cut -f3 -d';'` # obter a especiali.
        set nf = `grep $idf funcionarios.dat | cut -f2 -d';'` #obter o nome fun.
        # mostrar data - doente - medico : especialidade - valor
        echo $x | sed -e "s/$idf/$nf/" | sed -e "s/$idm/$nm;$esp/" | awk -F";"
    '{printf $4" - "$1" - "$2" : "$3" - "$5\n"}'
        @ total = $total + `echo $x | awk -F";" '{printf $4}'` # calc total
    end
    echo " "
    echo " Total = $total " # mostra total
    breaksw
endsw

# -----
case "7" :
    echo "7 Remover Ficha de Funcionario"
    echo " "
    echo "Introduza o nome do Funcionario :>"
    set nf = $<
    set nf = `echo $nf | tr "[a-z]" "[A-Z]"`
        set em = `grep $nf funcionarios.dat | wc -l` # verifica se existe
    if ($em > 0) then
        set idf = `grep $nf funcionarios.dat | cut -f1 -d';'` #obtem id de func.
        # remove todas as linha do ficheiro que contenham o id
        grep -v $idf funcionarios.dat > funcionarios.dat # no ficheiro funcionarios
        grep -v $idf relaciona.dat > relaciona.dat # no ficheiro relaciona
    endif
    breaksw

# -----
case "8" : # mesma funcionalidade do remover funcionario
    echo "8 Remover Ficha de Medico"
    echo " "
    echo "Introduza o nome do medico :>"
    set nm = $<
    set nm = `echo $nm | tr "[a-z]" "[A-Z]"`
    set em = `grep $nm medicos.dat | wc -l`
    if ($em > 0) then
        set idm = `grep $nm medicos.dat | cut -f1 -d';'`
        grep -v $idm medicos.dat > medicos.dat
        grep -v $idm relaciona.dat > relaciona.dat
    endif
    breaksw

```

```
# -----  
case "9":  
    breaksw  
else :  
    echo "Opcao invalida !"  
    breaksw  
endsw  
# -----  
if ($opt != 9) then  
    echo "(prima <ENTER> para continuar)"  
    set xpto = $<  
endif  
  
end  
  
# FIM DO SCRIPT
```

```
#!/bin/csh
#-----
# Autor      : Jose Nicolau (5205) & Tiago Pina (5223)
#             IPT - EI/SO 2003/2004
#
# Uso        : trb2 <path>
#             path - directorio onde iniciar a pesquisa de links ftp e http
#
# Output     : allink.html
#
# Descricao  : procura recursivamente numa determinada arvore de directorias
#             ficheiros *.htm* e constroi uma tabela HTML com os links
#             contidos nesses ficheiros (links http e ftp)
#-----

#-----
# Remove os ficheiros temporarios
if (-e files.txt) then
  rm files.txt
endif

if (-e links.txt) then
  rm links.txt
endif

if (-e allink.html) then
  rm allink.html
endif

echo "Script CSH trb2"
echo "jose nicolau (5205) e tiago pina (5223)"
echo "-----"
# -----
# Verificar se existe o parametro <path> se sim processa
if ($1 == '') then
  echo "USO  trb2 <path>"
  echo "Ex.  ./trb2 /home/www/public_html"
else
# -----
# Obter o parametro <path> e construir a path de pesquisa
set mypath = "$1/*.htm*"
echo "a processar: $mypath"

# -----
# Criar um ficheiro temporario com todos nome dos ficheiros .htm*
ls -lRa $mypath | awk '{print $9}' >> files.txt

# -----
#Criar o header do ficheiro HTML
set dt = `cat files.txt`
echo "<html><head><title>CSHELL Script - allink</title></head><body>" >> allink.html
echo "<table border=1 bordercolor=#000000 width=100%><tr><td><center>" >> allink.html
echo "<b>`date +%d-%m-%Y` ---- allink.html ---- " >> allink.html
echo "`date +%X`</b></center></td></tr>" >> allink.html
set cnt = 1

# -----
# Por cada ficheiro existente (.htm*) construir uma linha de header
foreach x ($dt)
  echo "<tr><td><center>$cnt -- $x</center></td></tr><tr><td>" >> allink.html
```

```
# -----
# Obter todas as linhas com links e substituir todos os
# caracteres ' >< por (NewLine) \n
grep http:// $x | tr '><' '\n' | tr "'<>' '\n' > links.txt
grep ftp:// $x | tr '><' '\n' | tr "'<>' '\n' >> links.txt

# -----
# Criar uma lista dos links existentes no ficheiro .htm*
set lnk = `grep http:// links.txt`

# -----
# Construir as linhas da tabela com base nos link existentes
foreach l ($lnk)
  echo "<a href='$l'>$l<a><br>" >> allink.html
end
set lnk = `grep ftp:// links.txt`
foreach l ($lnk)
  echo "<a href='$l'>$l<a><br>" >> allink.html
end
echo "</td></tr>" >> allink.html
@ cnt = $cnt + 1
end

# -----
# Terminar o corpo do ficheiro HTML
echo "</table></body></html>" >> allink.html

echo "processo concluido..."
endif

# FIM DO SCRIPT
```