

MATEMÁTICA / INFORMÁTICA
ENSINO da MATEMÁTICA
ENSINO da INFORMÁTICA

PROGRAMAÇÃO I

- Folha prática 1

Primeiro contacto com o sistema operativo VAX/VMS

Existe uma rede de computadores na UBI. Cada computador constitui um nó da rede e é identificado por um nome. O computador em que vai trabalhar é um VAX, e é designado por ALPHA.

→ Entrada no sistema

Após ligar o terminal surgirá um indicador ("prompt") do tipo:

Local UBI1>

Para estabelecer ligação com a máquina ALPHA dar o comando:

Local UBI1> **connect ALPHA**

e premir a tecla **<RETURN>** .

O sistema começa por lhe perguntar o seu:

Username :

Introduzir o nome de utilizador que lhe foi atribuído seguido da tecla **<RETURN>**.

O sistema responde perguntando-lhe a sua:

Password :

Introduzir a palavra de passe que lhe foi atribuída (que não verá) e premir a tecla **<RETURN>**.

Se se enganou em alguma coisa voltar a carregar em **<RETURN>** e recomeçar a sequência de comandos.

→ Mudar a Password

- Quando entrar pela primeira vez na sua área de trabalho, vai receber um aviso dizendo que deve mudar a password. Após digitar cada linha de informação, deverá submeter essa informação ao sistema operativo premindo a tecla **<RETURN>**.

New password: **"password nova" <RET>**

Verification: **"password nova" <RET>**

§

Quando quiser mudar novamente a sua password poderá fazê-lo dando o seguinte comando:

```
$ set password <RET>
Old password: "password antiga" <RET>
New password: "password nova" <RET>
Verification: "password nova" <RET>
$
```

→ Sair do sistema

Experimente sair do sistema com o comando,

```
$ logout <RET>
```

e entrar de novo com a nova password.

→ Criar ficheiros

Criar um ficheiro de nome EXEMPLO.TXT dando o comando :

```
$ edit exemplo.txt <RET>
```

Introduzir um texto à sua escolha. Quando terminar indique ao editor que chegou ao fim do ficheiro premindo simultaneamente as teclas <Ctrl> e <Z>.

Verifique que criou o ficheiro dando o comando:

```
$ directory <RET>
```

Para rever o seu conteúdo dar o comando:

```
$ type exemplo.txt <RET>
```

Copie o conteúdo deste ficheiro para um outro de nome EXEMPLO_DOIS.TXT, dando o comando:

```
$ copy <RET>
_From: exemplo.txt <RET>
_To: exemplo_dois.txt <RET>
$
```

Modifique agora o nome ao ficheiro criado (exemplo_dois.txt) para EXEMPLO_TRES.TXT, dando o comando:

```
$ rename <RET>
_From: exemplo_dois.txt <RET>
```

```
_To: exemplo_tres.txt <RET>  
$
```

Verificar o sucesso da operação com o comando: \$ **directory** <RET>

Eliminar o ficheiro EXEMPLO.TXT com o comando :

```
$ delete exemplo.txt;1 <RET>
```

Verificar o sucesso da operação com o comando: \$ **directory** <RET>

Criar uma nova versão do ficheiro EXEMPLO_TRES.TXT com o comando:

```
$ edit exemplo_tres.txt <RET>
```

Altere o ficheiro escrevendo um texto à sua escolha..

Pode explorar os comandos do editor premindo a tecla <PF4>. Aparecerá ao fundo do ecrã uma linha de comando:

Command:

- Se der o comando **HELP**, obterá informação sobre os vários comandos disponíveis.

Quando terminar indique que chegou ao fim premindo <Ctrl><Z>.

Verificar que foi criada uma segunda versão com o comando:

```
$ directory <RET>
```

→ **Procedimentos de comandos**

A linguagem de comandos DCL (Digital Command Language) é a linguagem que utilizamos para enviar directivas ao sistema operativo.

Um procedimento de comandos é um ficheiro com comandos DCL, cuja extensão por omissão (“default”) é **COM**, e que será executado com o comando, **@ nome_do_ficheiro**.

Como exemplo crie o ficheiro, COMANDOS.COM, com os dois comandos seguintes:

```
$ dir /date/size
```

```
$ show time
```

(Nota : Cada linha de um ficheiro de comandos deve começar com o carácter \$).

Depois de gravar o ficheiro, vai executá-lo com o comando:

\$ @ comandos

Verificará que foram executados os dois comandos do procedimento.

→ Símbolos

Há comandos que por serem muito extensos e/ou serem utilizados frequentemente, interessa simplificar a sua utilização recorrendo à definição de símbolos. Um símbolo é um nome simbólico com 1 a 255 caracteres alfanuméricos, dos quais o primeiro terá que ser alfabético.

O comando

\$ LI*MPA ::= SET TERMINAL / WIDTH = 80

cria o símbolo LIMPA, de abreviatura mínima LI. Uma vez definido o símbolo LIMPA, ele valerá pelo seu conteúdo e como tal poderá ser utilizado para substituir o comando SET TERMINAL / WIDTH = 80.

A definição do símbolo existirá durante toda a sua sessão de trabalho. Sempre que entrar na sua conta, poderá redefini-lo novamente. Se possuir vários símbolos, essa tarefa poderá ser simplificada recorrendo a um ficheiro com comandos.

O ficheiro LOGIN.COM é um ficheiro de comandos que, se existir na directoria principal é executado automaticamente no início de cada sessão de trabalho.

Crie um ficheiro LOGIN.COM , na sua directoria principal, com a definição do símbolo LI*MPA dada acima.

Execute o ficheiro com o comando:

\$ @login

→ Directorias

A vantagem de uma estrutura de directorias reside na possibilidade de se agruparem ficheiros por afinidades de acordo com o critério de cada um.

Suponhamos que quer agrupar todos os programas desta cadeira e à parte os restantes trabalhos:

Teria a seguinte estrutura:

(Nota: Onde está username deverá ler o **seu** nome de utilizador.)

USERNAME

PROGRAMACAO

OUTRAS

A especificação das directorias seria :
[USERNAME.PROGRAMACAO] e
[USERNAME.OUTRAS]

- Para criar a directoria que irá agrupar os seus trabalhos de Programação dê o comando:

\$ CREATE / DIRECTORY [USERNAME.PROGRAMACAO]

- Para criar a outra directoria :

\$ CREATE / DIRECTORY [USERNAME.OUTRAS]

- Transitar de directoria corresponde a alterar a especificação por omissão que o sistema junta aos nomes dos ficheiros que vai criando.

- Posicione-se na directoria PROGRAMACAO dando o comando:

\$ SET DEFAULT [USERNAME. PROGRAMACAO] (1)

- Para verificar que está na directoria pretendida dê o comando:

\$ SHOW DEFAULT

- Posicione-se na directoria OUTRAS dando o comando:

\$ SET DEFAULT [USERNAME.OUTRAS]

- Transite para a directoria principal dando o comando:

\$ SET DEFAULT [USERNAME]

Para transitar de directoria é possível fazer a especificação das directorias de modo simplificado. Isso consegue-se recorrendo às seguintes abreviaturas:

O carácter “.” significa “directoria actual”

O carácter “—” significa “directoria de nível anterior”

Por exemplo se estiver na sua directoria principal, [USERNAME], e quiser ir para a subdirectoria PROGRAMACAO, pode abreviar o comando (1) por:

\$ SET DEFAULT [. PROGRAMACAO]

Depois de verificar se está na directoria “certa” pode regressar novamente à directoria principal com o comando:

\$ SET DEFAULT [-]

Um problema do comando set default, é o facto de não assinalar um erro quando o utilizador estabelece como especificação por omissão um nome errado, isto é, um nome que não corresponde a uma directoria existente. Sempre que estiver “perdido”, sem saber onde está posicionado na sua área de trabalho, pode regressar à sua directoria principal com o comando,

\$ set default SYS\$LOGIN

Para facilitar a "movimentação" entre directorias podemos definir símbolos que simplifiquem os comandos anteriores.

Adicione ao seu ficheiro **login.com** a seguinte linha de comando:

\$ PROG ::= SET DEF [USERNAME.PROGRAMACAO]

Verifique que o símbolo PROG lhe permite colocar-se na directoria correspondente.

→ O comando HELP

O comando mais importante do sistema operativo com que vai trabalhar é o comando **help**, onde pode obter a descrição, o formato e alguns exemplos de todos os comandos disponíveis. Experimente-o e procure descobrir como apagar uma directoria...