

Programação II

Guião das Aulas Práticas

Departamento de Electrónica, Telecomunicações e Informática
Universidade de Aveiro

2016–2017

Resumo

Este guião propõe problemas para serem resolvidos pelos alunos de Programação II durante as respectivas aulas práticas e fora delas.

Aula Prática 0

Introdução ao UNIX

Resumo:

- Introdução ao sistema operativo UNIX.
- O ambiente de trabalho para as aulas práticas.
- Edição, compilação e execução de programas em Java.

As aulas práticas de Programação II decorrem em salas equipadas com computadores pessoais correndo o sistema operativo Linux. O Linux (ou mais correctamente, GNU/Linux) é uma variante *livre* e *gratuita* do conhecido sistema operativo UNIX. O número de instalações deste sistema tem registado um crescimento impressionante nos últimos anos, sendo actualmente um dos sistemas operativos mais populares em computadores pessoais, juntamente com o Windows e o MAC OS.¹ Dada a sua arquitectura aberta, existem inúmeras variantes de Linux adaptadas a diferentes utilizações que correm numa grande diversidade de máquinas desde computadores de bolso a servidores de Internet ou supercomputadores para cálculo científico. Na Universidade de Aveiro, um grupo de utilizadores de Linux denominado GLUA² disponibiliza diversas distribuições populares de Linux e organiza sessões de esclarecimento e de ajuda para quem estiver interessado em instalar e utilizar este sistema.

0.1 O Arranque, *Login* e *Logout*

Os computadores das salas de aula têm actualmente dois sistemas operativos instalados: o Windows e o Linux.

Assim, ao ligar o computador será confrontado com um menu para escolher o sistema que deseja iniciar. Terá alguns segundos para escolher a opção certa (o Linux, neste caso), usando as teclas de direcção ↑ ou ↓ e a tecla **Enter** ↵. Se o computador já se encontrar ligado e a correr Windows, deverá seleccionar a opção para reiniciar e poder voltar ao menu de arranque.

¹As últimas versões do MAC OS também implementam uma variante de UNIX.

²<http://glua.ua.pt>

Logo que o sistema esteja em funcionamento, aparece um ecrã de boas-vindas onde terá de se identificar, introduzindo o *nome-de-utilizador* (*username*) do tipo **a12345** (sem **@ua.pt**), e a *palavra-passe* (*password*) correspondente. Estes dados são os mesmos que utiliza para aceder ao ambiente Windows. Se introduziu os dados correctos, surge um ambiente gráfico que lhe permite interagir com o sistema e completar os exercícios da aula. Chama-se *entrar no sistema* (em Inglês *log in* usualmente escrito *login*) a este processo de autenticação para ter acesso ao sistema.

Quando terminar de usar o sistema, deve sempre *sair do sistema* (*log out* ou *logout*) de forma a que mais ninguém tenha acesso à sua área de trabalho. Se quiser desligar ou reiniciar o computador deve escolher a acção desejada no ecrã de boas-vindas que entretanto reaparece.

Exercício 0.1

Entre no sistema, introduzindo o seu nome-de-utilizador e palavra-chave na janela de *login*. Explore os menus e ícones do ambiente gráfico. Descubra a opção de *Log Out* (geralmente *System/Quit/Log Out*) e seleccione-a para sair do sistema. Repita o processo de *login* para regressar ao sistema.

0.2 A Linha de Comandos UNIX

Quando o sistema UNIX foi concebido, os computadores eram controlados essencialmente através de *consolas* ou *terminais* de texto: dispositivos dotados de um teclado e de um ecrã onde se podia visualizar somente texto. A interacção com o sistema fazia-se tipicamente através da introdução de comandos escritos no teclado e da observação da resposta produzida no ecrã pelos programas executados. Actualmente existem ambientes gráficos que correm sobre o UNIX e permitem visualizar informação de texto e gráfica, e interagir por manipulação virtual de objectos gráficos recorrendo a um rato e ao teclado. É o caso do Sistema de Janelas X, ou simplesmente X, que está instalado nos computadores das salas de aula. Apesar das novas formas de interacção proporcionadas pelos ambientes gráficos, continua a ser possível e em certos casos preferível usar a interface de *linha de comandos* para muitas operações. No X, isto pode fazer-se usando um *emulador de terminal*, um programa que abre uma janela onde se podem introduzir comandos linha-a-linha e observar as respostas geradas tal como num terminal de texto à moda antiga.

Exercício 0.2

Abra uma janela de terminal (a partir do menu principal)³ e quando surgir o *prompt*⁴ execute o comando **date**.

Observe que a resposta foi impressa imediatamente a seguir à linha do comando, de forma concisa, sem distrações nem grandes explicações. Este comportamento é usual

³Possivelmente: *Applications/Accessories/Terminal*.

⁴<http://pt.wikipedia.org/wiki/Prompt>

em muitos comandos UNIX e é típico de um certo estilo defendido pelos criadores deste sistema. Simples, mas eficaz.

Exercício 0.3

Execute o comando `cal` e observe o resultado. Descubra em que dia da semana nasceu, passando o mês e o ano como *argumentos* ao comando `cal`, por exemplo: `cal jan 1981`.

Os comandos em UNIX têm sempre a forma:

```
comando argumento1 argumento2 ...
```

onde **comando** é o nome do programa a executar e os argumentos são cadeias de caracteres, que podem ser incluídas ou não, de acordo com a sintaxe esperada por esse programa.

Na linha de comandos é possível recapitular um comando dado anteriormente usando as teclas de direcção ↑ e ↓. É possível depois editá-lo para produzir um novo comando com argumentos diferentes, por exemplo. Outra funcionalidade muito útil é a possibilidade de o sistema completar automaticamente comandos ou argumentos parcialmente escritos usando a tecla `Tab`.

0.2.1 Navegação no Sistema de Ficheiros

Tal como noutros sistemas operativos, no UNIX a informação é armazenada numa estrutura hierárquica formada por directórios, subdirectórios e ficheiros. O directório-raiz desta árvore é representado simplesmente por uma barra “/”. Cada utilizador possui um directório próprio nesta árvore, a partir do qual pode (e deve) criar e gerir toda a sua sub-árvore de directórios e ficheiros: é o chamado *directório do utilizador* ou *home directory*. Após a operação de *login* o sistema coloca-se nesse directório. Portanto neste momento deve ser esse o *directório actual* (*current directory*). Para saber qual é o directório actual execute o comando `pwd`. Deve surgir um nome como

```
/homermt/a12345
```

que indica que está no directório `a12345` que é um subdirectório de `homermt` que é um subdirectório directo da raiz `/`. Para listar o conteúdo do directório actual execute o comando `ls`. Deve ver uma lista dos ficheiros (e subdirectórios) contidos no seu directório neste momento, por exemplo:

```
arca Desktop Examples
```

Neste caso, observam-se dois subdirectórios e um *soft link* que é um tipo de ficheiro especial que serve de atalho para outro ficheiro ou directório. Dependendo da configuração do sistema, os nomes nesta listagem poderão aparecer com cores diferentes e/ou com uns caracteres especiais (`/`, `@`, `*`) no final, que servem para indicar o tipo de ficheiro mas de facto não fazem parte do seu nome. (Num ambiente gráfico a mesma informação está disponível numa representação mais visual. Experimente, por exemplo, escolher *Places/Home Folder* para ver o conteúdo do seu directório pessoal.)

Ficheiros cujos nomes começam por “.” não são listados por defeito, são ficheiros *escondidos*, usados geralmente para guardar informações de configuração de diversos programas. Para listar todos os ficheiros de um directório, incluindo os escondidos, deve executar a variante **ls -a**.

Por vezes é necessário listar alguns atributos dos ficheiros para além do nome. Pode fazê-lo executando as variantes **ls -l** ou **ls -la**.

```
total 88
drwx----- 13 a12345 users 4096 2007-01-26 14:03 .
drwxr-xr-x   3 root  root  4096 2007-01-25 10:52 ..
drwx-----  1 a12345 users    0 2007-01-26 08:00 arca
drwxr-xr-x   2 a12345 users 4096 2007-01-25 10:52 Desktop
lrwxrwxrwx   1 a12345 users  26 2007-01-25 10:52 Examples -> ...
```

Os principais atributos mostrados nestas listagens longas são:

Tipo de ficheiro identificado pelo primeiro carácter à esquerda, sendo **d** para directório, **-** para ficheiro normal, **l** para *soft link*, etc.

Permissões representadas por 3 conjuntos de 3 caracteres. Indicam as permissões de leitura **r**, escrita **w** e execução/pesquisa **x** relativamente ao dono do ficheiro, aos outros elementos do mesmo grupo e aos restantes utilizadores da máquina.

Propriedade indica a que utilizador e a que grupo pertence o ficheiro.

Tamanho em número de bytes.

Data e hora da última modificação.

Nome do ficheiro.

Normalmente existe um *alias*⁵ **ll** equivalente ao comando **ls -l**.

Além do **ls** e variantes, existem outros comandos importantes para a observação e manipulação de directórios, por exemplo:

cd — o directório actual passa a ser o directório do utilizador.

cd dir — o directório actual passa a ser o directório **dir**.

mkdir dir — cria um novo directório chamado **dir**.

rmdir dir — remove o directório **dir**, desde que esteja vazio.

⁵Um *alias* é um nome alternativo usado em representação de um determinado comando. São criados usando o comando interno **alias**.

O argumento `dir` pode ser dado de uma forma absoluta ou relativa. Na forma absoluta, `dir` identifica o caminho (*path*) para o directório pretendido a partir da raiz de todo o sistema de ficheiros; tem a forma `/subdir1/.../subdirN`. Na forma relativa, `dir` indica o caminho para o directório pretendido a partir do directório actual; tem a forma `subdir1/.../subdirN`. Há dois nomes especiais para directórios: “.” e “..” que representam respectivamente o directório actual e o directório pai, ou seja, o directório ao qual o actual pertence.

Exercício 0.4

Execute os comandos seguintes e interprete os resultados:

```
ls -l /
cd /
pwd
ls -l
cd usr
ls
cd local/src
pwd
ls
cd ../../bin
ls
cd
pwd
```

Exercício 0.5

Experimente utilizar o programa gráfico gestor de ficheiros⁶ para navegar pelos mesmos directórios que no exercício anterior: `/`, `/usr`, `/usr/local/src`, etc.

Exercício 0.6

Mude o directório actual para o seu subdirectório `arca`. Liste o seu conteúdo. Reconhece algum dos ficheiros?

Importante: O subdirectório `arca` não é um directório local do PC onde está a trabalhar; é na verdade a sua área privada de armazenamento no Arquivo Central de Dados (ARCA⁷), um servidor de ficheiros da Universidade de Aveiro. Esta área também é acessível a partir do ambiente Windows e através da Web, e é natural que já aí tenha colocado ficheiros noutras ocasiões. É neste directório que deve gravar os ficheiros e directórios que criar no decurso das aulas práticas. Os computadores das salas de aulas foram programados para apagarem o directório de utilizador (e.g. `/homermt/a1245/`) sempre que são

⁶Acessível no menu *Places*.

⁷<https://arca.ua.pt>

reiniciados. Só o conteúdo do subdirectório **arca** é salvaguardado. É portanto aí que deve colocar todo o seu trabalho.

Exercício 0.7

Crie, no directório **arca**, um subdirectório chamado **prog2** e, dentro desse, um directório chamado **aula00**.

0.2.2 Manipulação de ficheiros

O Linux (UNIX) dispõe de diversos comandos de manipulação de ficheiros. Eis alguns:

cat fic — imprime no dispositivo de saída *standard* (por defeito o ecrã) o conteúdo do ficheiro **fic**.

rm fic — remove (apaga) o ficheiro **fic**.

mv fic1 fic2 — muda o nome do ficheiro **fic1** para **fic2**.

mv fic dir — move o ficheiro **fic** para dentro do directório **dir**.

cp fic1 fic2 — cria uma cópia do ficheiro **fic1** chamada **fic2**.

cp fic dir — cria uma cópia do ficheiro **fic** dentro do directório **dir**.

head fic — mostra as primeiras linhas do ficheiro de texto **fic**.

tail fic — mostra as últimas linhas do ficheiro de texto **fic**.

more fic — imprime no dispositivo de saída *standard* (por defeito o ecrã), página a página, o conteúdo do ficheiro **fic**.

grep padrão fic — selecciona as linhas do ficheiro texto **fic** que satisfazem o critério de selecção **padrão**.

wc fic — conta o número de linhas, palavras e caracteres do ficheiro **fic**.

sort fic — ordena as linhas do ficheiro **fic**.

find dir -name fic — procura um ficheiro com o nome **fic** a partir do directório **dir**.

Além destes pode ainda considerar outros tais como: **less**, **cut**, **paste**, **tr**, etc. Todos estes comandos podem ser invocados usando argumentos opcionais que configuram o seu modo de funcionamento.

Exercício 0.8

Transfira o ficheiro **Totoloto.java** da página da disciplina para o directório **aula00** que criou no exercício anterior. Imprima o seu conteúdo no ecrã. Experimente outros comandos da lista acima.

0.2.3 Ajuda *On-line*

O Linux dispõe de vários mecanismos de ajuda imediata para a maioria dos seus comandos. Dois dos mais importantes são acedidos através dos comandos `man` e `info`, sendo o primeiro comum em todos os sistemas UNIX e o segundo mais específico do projecto GNU. Muitos comandos aceitam também uma opção `--help` que apresenta um resumo da sua forma de utilização.

Por exemplo, para conhecer as muitas opções de execução do comando `ls` pode executar `man ls`, ou `info ls`, ou `ls --help`.

Nota: Para navegar ao longo das páginas apresentadas pelo `man` ou pelo `info` pode usar as teclas de direcção `↑`, `↓` ou as teclas `PageUp`, `PageDown`. Para abandonar as páginas de ajuda e regressar à linha de comando deve premir a tecla `q`. Estes programas têm outras possibilidades de navegação e pesquisa que poderá ficar a conhecer fazendo por exemplo, `man man` ou `info info`.⁸

0.3 Ambiente de Programação em Java

0.3.1 Edição

Comece por editar o programa `Totoloto.java`. Para esse efeito dispõe de vários editores de texto. Aconselhamos, no entanto, a usar o `geany`, o `gedit` (*Text Editor*) ou o `gvim` (*VI editor*), visto possuírem a função de realce da sintaxe da linguagem Java. Na janela de terminal pode usar o editor `vim`, embora este editor tenha uma aprendizagem mais difícil.⁹

0.3.2 Compilação e Execução

O ficheiro que acabou de editar é usualmente designado por programa fonte. O passo seguinte consiste em gerar um programa executável a partir do programa fonte. Isto é feito usando o comando:

```
javac Totoloto.java
```

Verifique se existem erros de compilação e corrija-os. Quando não houver erros, o `javac` gera um programa executável chamado `Totoloto.class`.

Para executar o novo programa, use o comando:

```
java Totoloto
```

⁸Pelo contrário, `busca busca`, `mata mata`, não têm qualquer significado conhecido em UNIX, mas pode sempre tentar!

⁹Também é possível lançar a partir da linha de comandos qualquer outro editor ou programa. Por exemplo, experimente o comando `geany &`.

0.3.3 Documentação

Por vezes, para compreender o significado e a forma de utilização de uma classe ou de uma função em Java temos que consultar a sua documentação. Por exemplo, já deve ter consultado a documentação da classe `String`¹⁰.

A linguagem Java fornece ferramentas para a extracção automática de documentação associada a programas. Assim, é possível manter a documentação nos próprios ficheiros de código fonte, dentro de comentários com a forma `/** ... */`.

Experimente utilizar o comando `javadoc` sobre o programa fonte fornecido.

```
javadoc -charset utf8 -d doc Totoloto.java
```

Abra o ficheiro `doc/index.html` com um *browser* à sua escolha (por exemplo o `firefox`). Compare a documentação gerada automaticamente com a estrutura e os comentários existentes no programa fonte. Experimente alterar os comentários e volte a gerar a documentação.

¹⁰<https://docs.oracle.com/javase/8/docs/api/java/lang/String.html>