

Introdução ao ambiente Linux para programadores de Python

João Rodrigues, António Neves, Artur Pereira

Fundamentos de Programação, 2019

Resumo:

- Introdução ao sistema operativo UNIX.
- A interface de linha de comando.
- Edição e execução de programas em Python.

As aulas práticas de Fundamentos de Programação decorrem em salas equipadas com computadores correndo o sistema operativo Linux. O Linux (ou mais corretamente, GNU/Linux) é uma variante *livre e gratuita* do sistema operativo UNIX. Apesar de não ser essencial para se programar em Python ou noutra linguagem, o sistema Linux é muito popular entre os programadores e outros utilizadores especializados.

Para frequentar a disciplina e estudar em casa não precisa de ter um sistema Linux. Pode simplesmente instalar o Python no sistema operativo que já conhece e usá-lo para testar os seus programas. Mas tem que se familiarizar com o sistema Linux da sala de aulas, porque será usado nas provas de avaliação. Nomeadamente, tem de:

- Saber abrir e usar um editor de texto ou um ambiente integrado de programação para escrever programas Python.
- Saber abrir um emulador de terminal e usar a linha de comandos para invocar o interpretador de Python em modo interativo ou para executar programas em Python.
- Conseguir navegar pelo sistema de ficheiros, quer em modo gráfico, quer em modo de texto.
- Saber procurar e executar programas utilitários como um browser de Internet e outros.

Se quiser aprender um pouco mais ou ganhar experiência de administração do sistema, pode aproveitar a oportunidade para instalar o Linux no seu computador. A próxima secção discute várias alternativas de instalação. Pode saltá-la, se não estiver interessado. As secções seguintes explicam como usar o Linux para editar e executar programas e fazer as tarefas mais comuns. Também há uma secção sobre a utilização da *linha de comandos* e sobre os comandos mais comuns.

1 Sobre a instalação de Linux

Existem várias formas de instalar um sistema Linux num computador que já tenha outro sistema operativo instalado. Estas formas diferem na complexidade da instalação, mas também na versatilidade e no desempenho durante a utilização.

Instalação *dual-boot*: O sistema Linux é instalado diretamente numa *partição* própria do disco, em paralelo com o sistema pré-existente. Sempre que liga o computador, o utilizador tem de escolher qual dos sistemas pretende arrancar. Quando arranca, o sistema tem acesso direto a todo o hardware e atinge o desempenho máximo, mas para comutar entre sistemas é preciso reiniciar o sistema. Esta é a forma mais tradicional de instalação, mas também a mais complexa porque é preciso “encolher” a partição do sistema pré-existente. Há sempre algum risco de corromper o disco e inutilizar o sistema nessa operação.

Instalação em máquina virtual: Nesta modalidade, o sistema linux é instalado numa *máquina virtual*, que é um computador simulado por um programa (chamado virtualizador) que corre dentro do sistema operativo pré-existente. A máquina virtual simulada tem de reservar alguns recursos do computador real e partilhá-los com os restantes programas do sistema hospedeiro. Isso penaliza o desempenho do sistema cliente e do sistema hospedeiro, mas pode-se comutar entre sistemas instantaneamente. A instalação não apresenta riscos e não é muito complexa, mas por vezes requer ajustes que variam de máquina para máquina.

Windows Subsystem for Linux (WSL): Nos sistemas com Windows 10 é agora possível instalar uma extensão oficial da Microsoft que permite instalar e correr programas de Linux diretamente dentro do sistema Windows. Não se trata de uma virtualização completa, a partilha de recursos é mais eficiente, mas não se fica com um sistema Linux puro. Apesar das limitações, permite ter uma linha de comandos funcional, com acesso a milhares de programas UNIX e tem uma instalação bastante simples. Instalando e configurando algum software extra também se consegue usar programas Linux em modo gráfico e não apenas em modo de texto.

Se tem um sistema Windows 10 e pretende usar Linux esporadicamente e especialmente em modo de texto, então o WSL é a solução mais aconselhável.

Se pretende usar e administrar um sistema Linux completo, mas vai continuar a usar principalmente o sistema atual, ou se tem necessidade de comutar frequentemente entre sistemas, então usar um sistema virtualizado é a melhor opção.

Se pretende usar e explorar o Linux mais a sério e com alguma frequência e não se importa de reiniciar o computador para usar outro sistema, então opte pela instalação dual-boot.

Na página da disciplina em elearning.ua.pt encontra instruções para instalar o WSL e instruções para instalar e usar Linux como *máquina virtual*. Se quiser instalar em dual-boot, recomendamos que siga as instruções oficiais de uma distribuição de Linux ou procure

ajuda, por exemplo do GLUA¹ (Grupo Linux da Universidade de Aveiro) que é um grupo formado por alunos da universidade, que disponibiliza diversas distribuições populares de Linux e organiza sessões de esclarecimento e de ajuda para quem estiver interessado em instalar e utilizar este sistema.

2 O Arranque, *Login* e *Logout*

Os computadores das salas de aula têm atualmente dois sistemas operativos instalados em dual-boot: o Windows e o Linux. Quando ligar o computador será confrontado com um menu para escolher o sistema que deseja iniciar. Terá alguns segundos para escolher a opção desejada usando as teclas de direção ↑ ou ↓ e a tecla **Enter** ↵. Se o computador já se encontrar ligado e a correr Windows, tem de o reiniciar para ter acesso ao menu de arranque.

Logo que o sistema esteja em funcionamento, aparece um ecrã de boas-vindas onde terá de se identificar, introduzindo um *nome-de-utilizador* (*username*) do tipo `ana.costa@ua.pt` (ou só `ana.costa`), e a *palavra-passe* (*password*) correspondente. Se introduzir os dados corretos, surge um ambiente gráfico que lhe permite interagir com o sistema. A este processo de autenticação para ter acesso ao sistema chama-se *entrar* no sistema ou *registar-se* no sistema. Em Inglês diz-se *log in*, usualmente escrito *login*.

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 1

Entre no sistema, introduzindo o seu nome-de-utilizador e palavra-chave na janela de *login*. Explore os vários elementos do ambiente gráfico. Descubra a opção de *Log Out* e seleccione-a para sair do sistema sem o encerrar nem reiniciar. Repita o processo de *login* para regressar ao sistema.

3 A Linha de Comandos UNIX

Quando o sistema UNIX foi concebido (*circa* 1970), os computadores eram controlados essencialmente através de *consolas* ou *terminais* de texto: dispositivos dotados de um teclado e de um ecrã, ligados por um cabo a um computador central e onde se podia visualizar somente texto. A interação com o sistema fazia-se introduzindo comandos escritos no teclado e observando a resposta produzida no ecrã pelos programas executados. Actualmente existem ambientes gráficos em UNIX que permitem visualizar informação de texto e gráfica, e interagir por manipulação virtual de objectos gráficos recorrendo a um rato, ao teclado ou a outros dispositivos.

¹<http://glua.ua.pt>

Apesar das novas formas de interação proporcionadas pelos ambientes gráficos, continua a ser possível e por vezes preferível usar uma *interface de linha de comandos* para certas operações. Num ambiente gráfico, 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 2

Abra uma janela de terminal, clicando no ícone correspondente ou pressionando a tecla *Super* (geralmente tem o símbolo do Windows ou do Mac) e escrevendo **terminal** na caixa de procura que surge. Quando surgir o *prompt*² da linha de comando, 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 em muitos comandos UNIX e é típico de um certo estilo defendido pelos criadores deste sistema. Simples, mas eficaz.

Exercício 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 2001**.

Os comandos em UNIX têm sempre a forma:

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

onde **comando** é o nome do programa a executar e os argumentos são sequências de caracteres, separadas por espaços, 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 direção ↑ e ↓. É possível depois modificá-lo para produzir um novo comando com argumentos diferentes, por exemplo. Outra funcionalidade muito útil é a capacidade de o sistema completar automaticamente comandos ou argumentos parcialmente escritos, quando se pressiona a tecla **Tab**.

3.1 Navegação no Sistema de Ficheiros

Tal como noutros sistemas operativos, no UNIX a informação é armazenada numa estrutura hierárquica formada por diretórios, subdiretórios e ficheiros. O diretório-raiz desta árvore é representado simplesmente por uma barra “/”. Cada utilizador possui um diretório próprio nesta árvore, a partir do qual pode (e deve) criar e gerir toda a sua sub-árvore de diretórios e ficheiros: é o chamado *diretório pessoal* ou *home directory*. Quando faz o *login*, o sistema coloca-se nesse diretório. Portanto neste momento deve ser esse o *diretório actual* (*current directory*).

²<http://pt.wikipedia.org/wiki/Prompt>

Para saber qual é o diretório actual, execute o comando **pwd**. Deve surgir um nome como

```
/home/ana.costa
```

que indica que está no diretório **ana.costa** que é um subdiretório de **home** que é um subdiretório directo da raiz **/**.

Para listar o conteúdo do diretório actual execute o comando **ls**. Deve ver uma lista dos ficheiros (e subdiretórios) contidos no seu diretório neste momento, por exemplo:

```
arca Desktop Examples
```

Neste caso, observam-se dois subdiretórios e um *soft link* que é um tipo de ficheiro especial que serve de atalho para outro ficheiro ou diretório. 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, abrir o navegador de ficheiros e escolher *Home* para ver o conteúdo do seu diretório pessoal.

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

Por vezes é necessário listar alguns atributos dos ficheiros para além do nome. Para isso, use as variantes **ls -l** ou **ls -la**.

```
total 88
drwx----- 13 ana.costa users 4096 2007-01-26 14:03 .
drwxr-xr-x  3 root      root  4096 2007-01-25 10:52 ..
drwx-----  1 ana.costa users    0 2007-01-26 08:00 arca
drwxr-xr-x  2 ana.costa users 4096 2007-01-25 10:52 Desktop
lrwxrwxrwx  1 ana.costa 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 diretório, **-** 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* `ll` equivalente ao comando `ls -l`.³

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

`cd dir` — muda o diretório actual para `dir`.

`cd` — muda o diretório actual de volta ao diretório pessoal.

`mkdir dir` — cria um novo diretório chamado `dir`.

`rmdir dir` — remove o diretório `dir`, desde que esteja vazio.

O argumento `dir` pode ser dado de uma forma absoluta ou relativa. Na forma absoluta, `dir` identifica o caminho (*path*) para o diretório 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 diretório pretendido a partir do diretório actual; tem a forma `subdir1/.../subdirN`.

Em cada diretório há sempre dois nomes especiais: “.” e “..” que representam respectivamente o próprio diretório e o diretório pai, ou seja, o diretório ao qual este pertence.

Exercício 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
```

³Um *alias* é um nome alternativo usado em representação de um determinado comando. São criados usando o comando interno **alias**.

Exercício 5

Experimente utilizar o programa gráfico gestor de ficheiros⁴ para navegar pelos mesmos diretórios que no exercício anterior: `/`, `/usr`, `/usr/local/src`, etc.

Exercício 6

Mude o diretório actual para o seu subdiretório **arca**. Liste o seu conteúdo. Reconhece algum dos ficheiros? (O diretório **arca** é específico dos sistemas Linux instalados nas salas de aula na UA. Não existe noutros computadores.)

Importante: O subdiretório **arca** não é guardado no disco local do computador. É na verdade uma área privada de armazenamento no Arquivo Central de Dados (ARCA⁵), um servidor de ficheiros da Universidade de Aveiro. Esta área também é acessível a partir do ambiente Windows e através da Web, e pode ser que já contenha alguns ficheiros colocados lá noutras ocasiões. Normalmente é neste diretório que deve gravar os ficheiros e diretórios que criar no decurso das aulas práticas. Em alternativa, pode gravar num dispositivo de memória externo (USB flash drive).

Os computadores das salas de aulas foram programados para apagarem o diretório de utilizador (e.g. `/homermt/a1245/`) sempre que são reiniciados. Só o conteúdo do subdiretório **arca** é salvaguardado. É portanto aí que deve colocar todo o seu trabalho.

Exercício 7

Crie, no diretório **arca**, um subdiretório chamado **fp** e, dentro desse, um diretório chamado **aula01**.

3.2 Manipulação de ficheiros

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

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

`rm fic` — remove (apaga) o ficheiro **fic**. (Esta operação não pode ser desfeita!)

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

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

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

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

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

⁴Acessível no menu *Places*.

⁵<https://arcaweb.ua.pt>

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

`less fic` — mostra o conteúdo do ficheiro `fic`, página a página. Use as teclas de direção ↑, ↓, `PageUp` ou `PageDown` para navegar e `Q` para terminar.

`grep padrão fic` — selecciona as linhas do ficheiro `fic` que contêm texto indicado pelo 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 diretório `dir`.

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

Exercício 8

Descarregue o ficheiro com o material da aula e extraia o conteúdo para o diretório `aula01` que criou no exercício anterior. Imprima o seu conteúdo do ficheiro `primeiro.py` no ecrã. Experimente outros comandos da lista acima.

3.3 Obter ajuda

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 direção ↑, ↓ 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 descobrir nos manuais respetivos.

4 Ambiente de Programação em Python

4.1 Edição

Abra o programa `primeiro.py` com um editor de texto. O sistema Linux inclui vários editores de texto que pode usar. Aconselhamos, no entanto, a usar o `geany`, o `gedit` ou o `gvim` (*VI editor*), porque têm uma interface gráfica amigável e realce automático da sintaxe da linguagem Python (e outras). Também é possível lançar a partir da linha de comandos qualquer outro editor ou programa. Por exemplo, experimente o comando `geany &`.

4.2 Execução

O ficheiro que acabou de editar é usualmente designado por programa fonte. O passo seguinte consiste em *executar* o programa desenvolvido. Isto é feito usando o comando `python3 primeiro.py` no terminal. Se não houver erros, o resultado da execução do programa aparece no terminal.

Caso existam erros de sintaxe ou de execução, serão apresentados pela ordem em que foram detectados. Nesse caso, deverá tentar identificar a causa do primeiro erro e voltar a editar o programa para o corrigir.