

Terminal de Linux

Tópicos de Informática para Automação

Mário Antunes

September 22, 2025

Exercícios

Exercício 1: A Orientar-se ☐

Este exercício abrange os comandos `pwd`, `ls`, `cd` e comandos básicos de informação.

1. Abra o seu terminal. Verifique a sua localização inicial (o seu diretório pessoal) imprimindo o diretório de trabalho atual. `bash $ pwd`
 2. Liste o conteúdo do seu diretório pessoal. De seguida, liste-o novamente mostrando **todos** os ficheiros no formato de lista **longa**. `bash $ ls $ ls -la`
 3. Navegue para o diretório de *logs* do sistema em `/var/log` e liste o seu conteúdo. `bash $ cd /var/log $ ls`
 4. Obtenha alguma informação: descubra o seu nome de utilizador e a data atual. `bash $ whoami $ date`
 5. Regresse ao seu diretório pessoal usando o atalho mais rápido. `bash $ cd ~`
-

Exercício 2: A Explorar Diretórios Chave do Sistema ☐

Reforce o seu conhecimento da estrutura do sistema de ficheiros visitando diretórios importantes do sistema.

1. Navegue para o diretório `/etc`, que contém ficheiros de configuração de todo o sistema. `bash $ cd /etc`
 2. Liste o seu conteúdo. Verá muitos ficheiros de configuração. `bash $ ls`
 3. Veja o conteúdo do ficheiro `os-release` para obter informação sobre a sua distribuição Linux. `bash $ cat os-release`
 4. Agora, navegue para o diretório `/bin` para ver onde muitos dos programas de comandos essenciais estão guardados. Liste o seu conteúdo e veja se reconhece algum. `bash $ cd /bin $ ls`
-

Exercício 3: A Criar e Gerir Ficheiros ☐

Neste exercício, irá criar, copiar, mover e apagar ficheiros e diretórios.

1. A partir do seu diretório pessoal, crie um novo diretório chamado TIA. `bash $ cd ~ $ mkdir TIA`
2. Navegue para dentro do seu novo diretório TIA. `bash $ cd TIA`
3. Crie um ficheiro vazio chamado `notes.txt`. `bash $ touch notes.txt`
4. Adicione texto ao seu ficheiro e depois veja o seu conteúdo. `bash $ echo "A minha primeira linha de texto." > notes.txt $ cat notes.txt`
5. Faça uma cópia do seu ficheiro com o nome `notes_backup.txt`. `bash $ cp notes.txt notes_backup.txt`
6. Renomeie `notes.txt` para `important_notes.txt`. `bash $ mv notes.txt important_notes.txt`
7. Faça a limpeza, apagando o ficheiro de *backup*. `bash $ rm notes_backup.txt`

Exercício 4: A Compreender Permissões □

Este exercício foca-se na leitura e alteração de permissões de ficheiros com o comando `chmod`.

1. Dentro do seu diretório `~/TIA`, crie um novo ficheiro chamado `secret_data.txt`. `bash` `$ touch secret_data.txt`
 2. Veja as permissões padrão do ficheiro. `bash` `$ ls -l secret_data.txt`
 3. Remova todas as permissões para toda a gente. `bash` `$ chmod 000 secret_data.txt`
 4. Tente ver o conteúdo do ficheiro. Deverá receber um erro de **"Permission denied"**. `bash` `$ cat secret_data.txt`
 5. Restaure a permissão de leitura e escrita **apenas para si**. `bash` `$ chmod u+rw secret_data.txt`
 6. Crie um ficheiro de *script* vazio `my_script.sh` e torne-o executável para si. Verifique as permissões depois para ver a alteração. `bash` `$ touch my_script.sh` `$ chmod u+x my_script.sh` `$ ls -l my_script.sh`
-

Exercício 5: A Encontrar Ficheiros e Conteúdo com `find` e `grep` □

Aprenda a localizar ficheiros por nome e a procurar por texto dentro deles.

1. Dentro de `~/TIA`, crie um subdiretório e um novo ficheiro dentro dele. `bash` `$ mkdir -p ~/TIA/reports` `$ echo "Este é um relatório confidencial." > ~/TIA/reports/report-2025.txt`
 2. Use o comando `find` para procurar qualquer ficheiro que termine em `.txt` dentro do seu diretório `TIA`. `bash` `$ find ~/TIA -name "*.txt"`
 3. Use o `grep` para procurar a palavra "confidencial" no seu novo ficheiro de relatório. A *flag* `-i` torna a pesquisa insensível a maiúsculas e minúsculas. `bash` `$ grep -i "confidencial" ~/TIA/reports/report-2025.txt`
-

Exercício 6: A Gerir Processos □

Aprenda a ver e a parar programas em execução a partir da linha de comandos.

1. Inicie um processo que irá correr em *background*. O comando `sleep` espera por um número específico de segundos, e o `&` envia-o para *background*. `bash` `$ sleep 120 &`
 2. Encontre o ID do Processo (PID) do comando `sleep`. Pode usar o `pgrep` para isto. `bash` `$ pgrep sleep`
 3. Agora, termine o processo usando o comando `kill` e o PID que acabou de encontrar. Substitua PID pelo número real do passo anterior. `bash` `$ kill PID`
 4. Verifique se o processo já não está a correr. O comando `pgrep sleep` agora não deverá devolver nada. `bash` `$ pgrep sleep`
-

Exercício 7: A Gerir Software com APT □

Vamos instalar e remover um programa usando o gestor de pacotes **APT**.

1. Primeiro, sincronize a lista de pacotes do seu sistema com os repositórios de *software*. `bash` `$ sudo apt update`
2. Procure por uma ferramenta de linha de comandos útil chamada `htop`. `bash` `$ apt search htop`
3. Agora, instale o `htop`. Terá de confirmar a instalação quando solicitado. `bash` `$ sudo apt install htop`
4. Execute o programa que acabou de instalar. Pressione `q` para sair. `bash` `$ htop`
5. Finalmente, faça a limpeza removendo o pacote do seu sistema. `bash` `$ sudo apt remove htop`

Exercício 8: A Combinar Comandos ☐

Vamos explorar o poder do **pipe** (|) e do **redirecionamento** (>>).

1. O comando `ps aux` lista todos os processos em execução. Use o *pipe* (|) para enviar este *output* para o `grep` para encontrar o seu próprio processo "bash".
`bash $ ps aux | grep "bash"`
 2. Crie um ficheiro de *log* com uma entrada. `bash $ echo "$(date): A iniciar o meu trabalho." > ~/TIA/activity.log`
 3. Use o operador de acréscimo (>>) para adicionar uma segunda linha ao ficheiro sem apagar a primeira. `bash $ echo "$(date): Terminado o exercício 8." >> ~/TIA/activity.log`
 4. Verifique se o seu ficheiro de *log* contém ambas as linhas. `bash $ cat ~/TIA/activity.log`
-

Exercício 9: A Personalizar o Seu Ambiente ☐

É hora de editar o seu ficheiro `.bashrc` para criar um atalho útil (um *alias*).

1. Abra o seu ficheiro `~/ .bashrc` usando o editor `nano`. `bash $ nano ~/ .bashrc`
 2. Vá até ao final do ficheiro e adicione a seguinte linha para criar um atalho `ll` para o comando `ls -alF`.
`bash alias ll='ls -alF'`
 3. Guarde o ficheiro e saia do `nano` (Ctrl+X, depois S, e de seguida Enter).
 4. Carregue as alterações na sua sessão atual. `bash $ source ~/ .bashrc`
 5. Teste o seu novo *alias*. `bash $ ll`
-

Exercício 10: A Compreender a Variável \$PATH ☐

Descubra como a *shell* encontra os comandos que executa.

1. Veja a variável `$PATH` atual. É uma lista de diretórios separados por dois pontos. `bash $ echo $PATH`
 2. Crie um *script* simples de uma linha no seu diretório `~/TIA` e torne-o executável. `bash $ echo '#!/bin/bash' > ~/TIA/hello $ echo 'echo "Olá do meu script personalizado!"' >> ~/TIA/hello $ chmod +x ~/TIA/hello`
 3. Tente executar o *script* pelo nome. Irá falhar porque não está num diretório listado na `$PATH`. `bash $ hello`
 4. Agora execute-o usando o seu caminho relativo. Isto funciona. `bash $./hello`
 5. Adicione temporariamente o seu diretório `~/TIA` à `$PATH`. Agora tente executar o *script* pelo nome novamente. `bash $ export PATH="$HOME/TIA:$PATH" $ hello` Esta alteração dura apenas para a sua sessão de terminal atual.
-

Exercício 11: Desafio de Scripting ☐

Vamos criar um *script* que automatiza a criação de uma estrutura de projeto.

1. Crie e abra um novo ficheiro chamado `setup_project.sh` no seu diretório `~/TIA`. Adicione o código seguinte, e depois guarde e feche o ficheiro.

```
#!/bin/bash
PROJECT_DIR="$HOME/TIA/my_project"

if [ -d "$PROJECT_DIR" ]; then
    echo "Erro: O diretório '$PROJECT_DIR' já existe."
    exit 1
fi

mkdir "$PROJECT_DIR"
```

```
echo "Diretório '$PROJECT_DIR' criado."
```

```
for folder in assets source docs
do
    mkdir "$PROJECT_DIR/$folder"
    echo "→ Subpasta criada: $folder"
done
```

```
echo "Configuração do projeto concluída!"
```

2. Torne o *script* executável e depois execute-o. `bash $ chmod +x ~/TIA/setup_project.sh`
`$ ~/TIA/setup_project.sh`
 3. Verifique se o diretório e os seus subdiretórios foram criados. `bash $ ls -R ~/TIA/my_project`
-

Exercício 12: A Agendar uma Tarefa com cron ☐

Vamos criar um *script* simples e agendá-lo para ser executado automaticamente a cada minuto.

1. **Crie o Script:** No seu diretório ~/TIA, crie um *script* chamado `log_time.sh` com o seguinte conteúdo. `bash #!/bin/bash date >> $HOME/TIA/cron_log.txt`
2. **Torne-o Executável:** `bash $ chmod +x ~/TIA/log_time.sh`
3. **Abra o seu Crontab:** Isto irá abrir um editor de texto. `bash $ crontab -e`
4. **Adicione o Cron Job:** Vá até ao final do ficheiro e adicione a seguinte linha. Deve usar o caminho completo e absoluto para o seu *script*. `cron * * * * * /home/student/TIA/log_time.sh`
5. **Guarde e Verifique:** Guarde e saia do editor. Espere dois minutos e depois verifique o seu ficheiro de *log*. Deverá ver duas entradas com data e hora. `bash $ cat ~/TIA/cron_log.txt`
6. **Limpeza:** É muito importante remover o *cron job* para que não corra para sempre. Este comando remove todo o seu ficheiro *crontab*. `bash $ crontab -r`