

Aula prática 1 – Guião de Familiarização com o Ambiente de Trabalho

Objectivo: Familiarização com o ambiente de desenvolvimento de programas (Ubuntu, Gedit, Terminal, Nautilus, C, GNU, gcc).

Nota: O aluno deve previamente obter no CICA a identificação (user ID/ login) e a palavra-chave (password).

1. Aceda à sua conta Linux (Ubuntu) fornecendo a sua identificação (*login*) e a palavra-chave (*password*).
2. Após o login, aparecerá o ambiente de trabalho do Ubuntu. Clique no menu principal (botão do canto superior esquerdo) e escreva ou procure pela aplicação “Gedit”. Ligue-a. Repita o procedimento para a aplicação “Terminal”.

Nota: Daqui em diante representamos por \$ os comandos a ser executados no Terminal.

3. Crie um diretório para os programas que fizer nesta aula, chamado “progl-tp1”:

```
$ mkdir progl-tp1
```

(“mkdir” significa make directory)

4. Utilize o seguinte comando para verificar o conteúdo do diretório corrente, certificando-se que o diretório “progl-tp1” foi criado:

```
$ ls
```

(“ls” significa list)

A palavra “progl-tp1” deve aparecer listada no ecrã.

5. Mude o diretório do Terminal para o diretório “progl-tp1”:

```
$ cd progl-tp1
```

(“cd” significa change directory)

Escreva de novo `ls` e verifique que o diretório “progl-tp1” está vazio (ou seja, que não aparece nada no ecrã aquando da execução do comando)

6. Na aplicação Gedit, escreva o seguinte programa em linguagem C:

```
#include <stdio.h>

main()
{
    printf("Good Morning!\n");
}
```

7. Guarde (*File* → *Save As*) o programa que editou, indicando o diretório do novo ficheiro (*Places: Home/prog1-tp1/*) e o respetivo nome (*Name: teste.c*).
8. Liste o conteúdo do diretório *prog1-tp1* no Terminal, verificando que nele já consta o ficheiro que acaba de ser criado.
9. Compile o programa que acabou de escrever em *teste.c*:

```
$ gcc teste.c
```

10. Execute o programa e verifique o resultado da execução:

```
$ ./a.out
```

No ecrã deve aparecer a frase “*Good Morning!*”. O ficheiro *a.out* é o nome por omissão do ficheiro executável criado pelo *gcc*.

11. Compile novamente o programa, desta vez especificando o nome do ficheiro executável:

```
$ gcc teste.c -o teste
```

A opção *-o* é uma abreviatura de *output*. O nome indicado depois dessa opção será o nome do *output*, ou seja, o nome do novo ficheiro executável.

12. Execute desta vez o ficheiro *teste* e verifique que se obtém o mesmo resultado que no passo 10.
13. Remova o primeiro programa que criou:

```
$ rm a.out
```

(*rm* significa *remove*; liste a diretoria em uso e verifique que *a.out* foi removido; **nota:** usar *rmdir* para remover diretórios vazios ou *rm -r* para remover diretórios não vazios)

14. Para evitar erros, no decorrer da escrita de um programa (como o *teste.c*) ou na execução de comandos no Terminal (como “*cd prog1-tp1*”) há muitas vezes a necessidade de verificar a sua sintaxe. Desta maneira, podemos obter informação sobre um comando através de *Man Pages* (Manuais de Informação) do próprio computador na aplicação Terminal.

Obtenha informação sobre o comando *rm*, escrevendo:

```
$ man rm
```

Navegue pelas instruções da página do manual do comando *rm* e confirme a funcionalidade do comando *rm -r*, mencionado no passo 13 e saia do manual.

(**Nota:** para sair do manual e regressar ao terminal, clique em “q” – significa “quit”)

15. Crie um novo programa com o seguinte código e guarde-o como *exp.c*:

```
#include <stdio.h>
main()
{
    int den, r
    den = 0;
    r = 10 % den;
    printf("%d\n", r);
}
```

16. Compile este novo programa.

```
$ gcc exp.c -o exp
```

Repare que um dos erros de compilação é:

```
`c:7: missing termination ` character'.
```

De facto, existe um erro sintático na linha 7.

17. No editor altere a linha 7 para:

```
printf("%d\n", r);
```

Guarde de novo o ficheiro com *Save*.

18. Compile de novo (no terminal, basta clicar na seta para cima e carregar em Enter).

O resultado desta vez deverá ser:

```
`c:6: 'r' undeclared'
```

De facto, na linha 4 falta um ";" a seguir à declaração da variável 'r';

19. No editor altere a linha 4 para:

```
int den, r;
```

Compilando de novo não dá erro mas, ao executar, aparece o erro "*floating point exception*".

20. Insira uma correção adequada e experimente de novo. Note que se não sabe a sintaxe de um comando que está a dar erro, talvez precise de a verificar.

21. Copie o ficheiro *teste.c* para *t.c*:

```
$ cp teste.c t.c
```

22. Liste o conteúdo do diretório *prog1-tp1*.

23. Copie um ficheiro do diretório atual (*prog1-tp1*) para um novo diretório dentro deste, com um nome à sua escolha. **Por exemplo:**

```
$ ls
$ mkdir pasta-de-teste
$ cp teste.c pasta-de-teste/
$ ls -R
```


24. Copie novamente o ficheiro para o mesmo diretório, mas com um outro nome.

Por exemplo:

```
$ cp teste.c pasta-de-teste/teste_copiado.c
$ ls -R
```

25. Termine a sessão de trabalho.

26. No final da sessão, desligue as diferentes janelas abertas (em *File*, selecione *Quit*)

Finalmente, no botão de desligar (canto superior direito ) , selecione *Logout*.