

Práctica 0

Preliminares de Linux y C++

Introducción a la computación

1^{er} cuatrimestre 2019

Ejercicio 1. Introduzca los siguientes comandos en una terminal de su cuenta en Linux e interprete los resultados.

```
1  whoami
2  date
3  pwd
4  cd ..
5  pwd
6  cd ..
7  pwd
8  cd
9  pwd
10 ls
11 mkdir dir1
12 ls
13 echo un poco de texto
14 echo un poco de texto > file1.txt
15 ls
16 cat file1.txt
17 file file1.txt
18 file dir1
19 ls
20 ls -l
21 cd dir1
22 pwd
23 cd ..
24 ls
25 clear
26 ls
27 rm file1.txt
28 ls
29 rmdir dir1
30 ls
31 exit
```

Ejercicio 2.

(a) Escriba el siguiente programa en un archivo `hola.cpp`:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(int argc, char* argv[]) {
6      cout << "Hola, mundo!\n";
7      return 0;
8  }
```

Compile el programa usando `g++`:

```
g++ -std=c++0x -o hola hola.cpp
```

Ejecute el programa y observe su salida:

```
./hola
```

(b) Reemplace la línea 4 del programa del punto (a) por la siguiente:

```
cout << "Hola, mundo!\n"
```

Compile el programa con g++ y observe la salida del compilador.

(c) Reemplace la línea 4 del programa del punto (a) por el siguiente fragmento de código:

```
cout << "Hola, mundo!"  
";
```

Compile el programa con g++ y observe la salida del compilador.

(d) Reemplace la línea 4 del programa del punto (a) por la siguiente:

```
cout << "Hola, mundo!";
```

Compile el programa con g++ y ejecútelo. ¿Qué diferencia encuentra con el programa original?

(e) Reemplace la línea 4 del programa del punto (a) por la siguiente:

```
cout << "\tHola, \tmundo!\t\n";
```

Compile el programa con g++ y ejecútelo. ¿Qué diferencia encuentra con el programa original?

(f) Reemplace la línea 4 del programa del ejercicio a por el siguiente fragmento de código:

```
/* Esto es un comentario */  
cout << "Hola, mundo!\n";
```

Compile el programa con g++ y ejecútelo. ¿Nota alguna diferencia con la salida del programa del punto (a)? ¿Qué se puede concluir del texto que figura entre “/*” y “*/” en el código fuente de un programa en C++?

Ejercicio 3. En el ejercicio 2 apareció el caracter ‘\’. Se conoce a la barra invertida como caracter de escape, y su función dentro de una cadena de caracteres es darle un significado especial al caracter que figura inmediatamente a continuación. Por ejemplo, ‘\n’ se usa como caracter de nueva línea.

(a) Escriba, compile y corra el siguiente programa

```
1  #include <iostream>  
2  
3  using namespace std;  
4  
5  int main(int argc, char* argv[]) {  
6      cout << "\\n";  
7      return 0;  
8  }
```

(b) Modifique la línea 4 del programa del punto (a) por la siguiente:

```
cout << "\\n";
```

Compile el programa con g++ y observe la salida del compilador.

(c) Escriba un programa en C++ que imprima por pantalla la secuencia “\n\n”.

Ejercicio 4.

(a) Escriba, compile y ejecute el siguiente programa:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(int argc, char* argv[]) {
6      cout << "El numero pi es aproximadamente 3.14\n";
7      return 0;
8  }
```

(b) Modifique la línea 4 del programa del punto (a) por la siguiente:

```
cout << "El numero pi es aproximadamente" << 3.14 << "\n";
```

Compile y ejecute el programa.

¿Cuál es la diferencia entre los programas de los items (a) y (b)?

Ejercicio 5.

(a) Escriba, compile y ejecute el siguiente programa:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(int argc, char* argv[]) {
6      float pi = 3.14;
7      pi = 3.141592;
8      cout << "Pi = " << pi << "\n";
9      return 0;
10 }
```

(b) Modifique la línea 4 del programa del punto (a) por la siguiente:

```
const float pi = 3.14;
```

Compile el programa con g++. ¿Cuál es el sentido de usar const?

Ejercicio 6. Escriba, compile y ejecute el siguiente programa:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main(int argc, char* argv[]) {
6      float e1 = 2.718;
7      int e2 = e1;
8      cout << "e1 = " << e1 << "\ne2 = " << e2 << "\n";
9      return 0;
10 }

```

¿A qué atribuye la diferencia entre e1 y e2?

Ejercicio 7.

Escriba el siguiente programa en un archivo `parametros.cpp`:

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main(int argc, char* argv[]) {
6      cout << "Cantidad de parámetros recibidos: " << argc-1 << "\n";
7      return 0;
8  }

```

Compile el programa usando `g++`:

```
g++ -std=c++0x -o parametros parametros.c
```

(a) Ejecute el programa mediante el siguiente comando:

```
./parametros
```

(b) Ejecute el programa mediante el siguiente comando:

```
./parametros hola
```

(c) Ejecute el programa mediante el siguiente comando:

```
./parametros cada palabra es un argumento distinto
```

(d) ¿Qué conclusión puede sacar del valor que tiene `argc` al comenzar la ejecución de `main`? (notar que el valor del argumento de `cout` es `argc-1`).

Ejercicio 8.

(a) Escriba el siguiente programa en un archivo `argumentos.cpp`:

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main(int argc, char* argv[]) {
7      string arg1 = argv[1];
8      string arg2 = argv[2];
9      string arg3 = argv[3];
10     string arg4 = argv[4];
11     cout << "Argumento 1 = " << arg1 << "\n";
12     cout << "Argumento 2 = " << arg2 << "\n";
13     cout << "Argumento 3 = " << arg3 << "\n";
14     cout << "Argumento 4 = " << arg4 << "\n";
15     return 0;
16 }
```

Compile el programa usando `g++`:

```
g++ -std=c++0x -o argumentos argumentos.cpp
```

(b) Ejecute el programa mediante el siguiente comando:

```
./argumentos argumentos separados por espacios
```

(c) ¿Qué conclusión puede sacarse sobre los valores que contiene el arreglo `argv`? ¿Cuál es la relación existente entre `argc` y `argv`?

(d) Escriba un programa que imprima por pantalla el contenido de `argv[0]`.

Ejercicio 9. Verifique si en C++ es posible que el nombre de una variable comience con:

- (a) un número;
- (b) el símbolo '#';
- (c) el símbolo '*';
- (d) el símbolo '_';

¿Puede ser "int" el nombre de una variable?