

Guión Práctica 1

José Díaz

24 de febrero de 2007

1. Introducción

En esta práctica aprenderemos los elementos de programación en C++ y de trabajo en un entorno UNIX. Leed atentamente el contenido del primer tema de Prácticas de Cálculo Numérico en C++, *practica1.pdf*.

2. Creación del directorio de trabajo

Conectarnos en un ordenador en Linux. Creamos un directorio de trabajo. Cuando entráis en Linux estáis en el directorio `/home/calnum`. Este directorio tiene como nombre abreviado `~`. Si cambiáis de directorio por alguna razón y queréis volver a este directorio, podéis hacerlo con el comando

```
cd ~
```

En el terminal alfanumérico (pantalla xterm) creamos un directorio con vuestro nombre (ej. mateo) escribiendo el siguiente comando:

```
mkdir mateo
```

El comando `mkdir` crea un directorio con el nombre que sigue. Este directorio existirá hasta que se borre el disco del ordenador, lo cual se hace pocas veces durante el curso.

Verificáis que se ha creado el directorio con el comando

```
ls
```

que lista todos los ficheros y directorios contenidos en el directorio donde se ejecuta el comando. El camino absoluto de este directorio es `/home/calnum/mateo` como podéis verificar escribiendo el comando

```
pwd
```

en el terminal alfanumérico.

3. Descarga y descompresión de la práctica

Con el Web browser (Mozilla), que encontráis en el menú del escritorio del ordenador `/Aplicaciones/Internet` vais a la página de Cálculo Numérico

`www.uv.es/~diazj/`
y en *Cálculo Numérico/Código de Prácticas/ Práctica 1* os bajáis el fichero comprimido del código de programas de la Práctica 1,

`P1.tgz`

Colocáis este fichero en `/home/calnum/mateo`. Si por casualidad no lo habéis descargado en este directorio, os situáis en el directorio donde lo habéis bajado y lo movéis con el comando

```
mv P1.tgz /home/calnum/mateo
```

Bajáis al directorio `/home/calnum/mateo`:

```
cd /home/calnum/mateo
```

y descomprimís el fichero `P1.tgz` con el comando:

```
tar -zxvf P1.tgz
```

Se creará el directorio

```
/home/calnum/mateo/P1
```

Bajáis al directorio `/home/calnum/mateo/P1` con

```
cd P1
```

Ya estáis en el directorio donde vais a realizar la práctica, y de donde en principio no tenéis por que salir hasta el final de la misma.

4. compilación y ejecución de los programas

Compiláis cada uno de los programas con el compilador `g++`. Por ejemplo, el fichero `hola.cc` lo compiláis con

```
g++ -o hola hola.cc
```

y lo ejecutáis escribiendo en la línea de comandos

```
./hola
```

Estudad el contenido de cada uno de los programas y realizad los ejercicios propuestos en el manual de la práctica 1, `practical.pdf`.

Una vez finalizados todos los ejercicios, ya estáis en condiciones de realizar la memoria.

5. Realización de la memoria

Para cada uno de los ejercicios pedidos en el último apartado de este guión, cread un fichero fuente que, por ejemplo, en el caso del programa del primer ejercicio llamaréis `ej1.cpp`. Cread igualmente los ficheros de datos en caso necesario, `ej1.dat`, por ejemplo. Compiláis el programa con

```
g++ -o ej1 ej1.cpp
```

y ejecutáis el programa en un fichero de salida `ej1.res`

```
./ej1<ej1.dat>ej1.res
```

Hacéis esto para todos los programas de los ejercicios.

Para confeccionar la memoria utilizáis Emacs, u otro editor ASCII, pero no MSWord o editores de RichText. Creáis un fichero llamado

```
P1_nombre_apellido1_apellido2.txt,
```

y con Emacs escribís vuestro nombre y grupo de prácticas (José Pérez García. Grupo BL3) en la primera línea e incluís en él todos los ficheros por orden correlativo: ej1.cpp, ej1.dat, ej1.res, ej2.cpp,etc. Escribís cualquier comentario a final de los ficheros de resultados del ejercicio correspondiente. Separáis cada ejercicio con una línea de caracteres, como por ejemplo,

```
=====
```

Imprimís este fichero, a ser posible a doble cara. Esta es la memoria que debéis presentar.

6. Antes de abandonar la sesión de prácticas

Os colocáis en `/home/calnum/mateo` con

```
cd ..
```

Hacéis un archivo comprimido de P1, con vuestro nombre, para que no se confunda con `P1.tgz` original.

```
tar -zcvf P1_mateo.tgz P1
```

Podéis verificar que están todos los ficheros, listando el archivo comprimido

```
tar -ztvf P1_mateo.tgz
```

Poneís este fichero en vuestro espacio web de la Universitat y lo salváis igualmente en disco o memoria USB. Tomad al menos dos medidas de precaución, para evitar pérdida de datos.

7. Entregar la memoria

El listado en papel de las práctica descrito en el apartado 5 lo entregáis en la siguiente sesión de prácticas, o en su defecto lo depositáis en la casilla de correos del profesor en el Dpto. de FAMN.

Depositáis el fichero `P1_nombre_apellido1_apellido2.txt` como respuesta a la Práctica 1 en el Aula Virtual de Cálculo Numérico.

8. Ejercicios a presentar como memoria

1. Escribir un programa para calcular $\cos(x)$ con 12 cifras decimales, a partir de su serie de potencias. Realizar una salida en fichero de los valores de $\cos(x)$ de 30 a 150 grados, de 30 en 30 grados. Imponer el límite de iteraciones a 1000 términos de la serie. El programa debe de imprimir en pantalla los resultados (con 12 cifras decimales) junto con el número de iteraciones necesarios para calcular cada valor. *Nota:* Utilizad el programa `while.cc`

como punto de partida donde introduciréis la cabecera `< iomanip >` para la salida de datos formateada. Notad que la serie del coseno es alterna y sólo tiene términos pares. Una forma de actualizar el término general es $t = -t * \left(\frac{x}{i}\right) * \left(\frac{x}{i+1}\right)$ con la que se evitan overflows y se tiene en cuenta el cambio de signo. El índice hay que actualizarlo como $i += 2$.

2. Realizar un programa para resolver una ecuación de segundo grado. Se deben de considerar cada uno de los casos del discriminante. La precisión requerida es 10^{-5} y los resultados se deben imprimir con 5 cifras decimales. Definir todos los reales del tipo `float`.
3. Realizar un programa que diga si un punto (x,y), leído mediante teclado, está comprendido en la zona limitada por los ejes coordenados y las líneas $y=x$ e $y=4$.
4. Verificad que la suma de N enteros es $N(N+1)/2$. ¿Hasta que número podéis verificarla declarando los enteros como 1) `int` y 2) `long int`? Tendréis que averiguar cuando el ordenador comienza a hacer tonterías.
5. Escribid un programa para calcular la función factorial. ¿Hasta que número funciona con `int` y con `long int`? ¿Se os ocurre una idea para extender su funcionamiento a números más grandes?
6. Dibujad con *Gnuplot* la función `sin(x)` entre 0 y 1 en trazo continuo, junto con un conjunto de 10 puntos igualmente espaciados de la función `x*sin(x)`. Para ello, realizad primero un programa que escriba en un fichero los pares `(x, x*sin(x))`, en forma de columnas que puedan ser leídas por *Gnuplot*.
7. Escribid un programa que dibuje una línea recta $y = ax + b$ con *Gnuplot*. El programa debe pedir la pendiente y la ordenada en el origen, el intervalo (x_{min}, x_{max}) en el que se desea dibujar la recta y el número de puntos en los que se desea calcular los valores de la recta. Debe igualmente preguntar el nombre del fichero en el que se deben de escribir los valores (se puede dar un nombre por defecto en caso de que no se introduzca nada, que se asignará cuando se teclee Ret sin introducir nada) y debe de escribir los valores tanto en pantalla cómo en el fichero. El fichero lo utilizaréis para dibujar las rectas con *Gnuplot*.
8. Repetid el ejercicio anterior para el caso de una parábola $y = ax^2 + bx + c$ y dibujad los polinomios de segundo grado utilizados para calcular sus raíces en el ejercicio 2.