

Guión Práctica 4

José Díaz

12 de marzo de 2007

1. Introducción

En esta práctica introduciremos nuevos conceptos de programación en C++ y de trabajo en un entorno UNIX. En particular estudiaremos punteros, clases, clases de vectores y matrices y las librerías `gnuplot_i++` para uso de `gnuplot` desde programas en C++ y la librería `mpfun` para el cálculo con aritmética de precisión arbitraria. Leed atentamente el contenido del cuarto tema de Prácticas de Cálculo Numérico en C++, `practica4.pdf`.

2. Creación del directorio de trabajo

Conectaros en un ordenador en Linux. Cread un directorio de trabajo con vuestro nombre, si no existe el que creasteis en prácticas anteriores. Cuando entráis en Linux estáis en el directorio `/home/calnum`. Si cambiáis de directorio por alguna razón y queréis volver, podéis llegar al mismo con el comando

```
cd ~
```

En el terminal alfanumérico (pantalla `xterm`) cread un directorio con vuestro nombre (ej. `cristina`) escribiendo el siguiente comando:

```
mkdir cristina
```

El comando `mkdir` crea un directorio con el nombre que sigue. Este directorio existirá hasta que se borre el disco del ordenador.

Verificáis que se ha creado el directorio con el comando

```
ls
```

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

```
pwd
```

en el terminal alfanumérico.

3. Descarga y descompresión de la práctica

Con el navegador Mozilla Web Browser vais a la página de Cálculo Numérico
`www.uv.es/~diazj/`

y os bajáis

P4.tgz

Descargáis este fichero en `/home/calnum/cristina`, y si accidentalmente lo descargáis en otro directorio, lo desplazáis con el comando

```
mv P4.tgz /home/calnum/cristina
```

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

```
cd /home/calnum/cristina
```

y descomprimís el fichero `P4.tgz` con el comando:

```
tar -zxvf P4.tgz
```

Se creará el directorio

```
/home/calnum/cristina/P4
```

Bajáis al directorio `/home/calnum/cristina/P4` con el comando

```
cd P4
```

Ya estáis en el directorio donde vais a realizar la práctica.

4. Compilación y ejecución de los programas

En el directorio `P4` se encuentra la librería `mpfun` en el archivo `mpfun`. La compilación de la librería se realiza con los comandos: `cd mpfun` y después `make`:

```
cd mpfun
```

y compiláis la librería :

```
make
```

Volvéis a `P4`:

```
cd ..
```

Aquí compiláis cada uno de los programas

```
g++ -o string string.cpp
```

y lo ejecutáis

```
./string
```

Los programas que utilizan la librería `mpfun` necesitan cabeceras que se encuentran en `mpfun/include` y la librería `mp` que se encuentra en `mpfun/lib`, por lo que la orden de compilación de `mpej1.cpp` debe escribirse como

```
g++ -I./mpfun/include -L./mpfun/lib -o mpej1 mpej1.cpp -lmp
```

Aparte del directorio `mpfun`, que acabáis de crear, se ha creado también el directorio `templates`, que contiene las cabeceras de las clases `TNT` y `JAMA`, y de la librería `gnuplot_i++`. La compilación de `tnt_ejemplos.cpp` que utiliza estas clases es:

```
g++ -I./templates -o tnt_ejemplos tnt_ejemplos.cpp
```

Los programas que utilizan cabeceras necesitan acceder a este directorio, por lo que se utiliza la opción `-I./templates` en el compilador. Finalmente, el programa

mpfun_tnt.cpp necesita tanto la librería mpfun como la TNT, por lo que la orden de compilación es

```
g++ -I./templates -I./mpfun/include -o mfun_tnt mfun_tnt.cpp -L./mpfun/lib
-lmp
```

Para utilizar las funciones de la librería TNT hay que incluir el namespace CALNUM en los programas mediante la sentencia:

```
using namespace CALNUM;
```

y también hay que incluir la cabecera algebralineal.h.

Para utilizar la librería gnuplot_i++ hay que incluir la cabecera gnuplot_i_calnum.h y compilar con gnuplot_i_calnum.cpp:

```
g++ -I./templates -o gnuploti_ejemplo gnuploti_ejemplo.cpp gnuplot_i_calnum.cpp
```

Estudiad el contenido de cada uno de los programas y realizad los ejercicios propuestos en el manual de la práctica 4, practica4.pdf.

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

5. Realización de la memoria

Para cada uno de los ejercicios pedidos abajo cread el fichero fuente que llamaréis p4ej1.cpp en el caso del programa del primer ejercicio, y los ficheros de datos en su caso, ej1.dat por ejemplo. Compiláis el programa

```
g++ -o p4ej1 p4ej1.cpp
y ejecutáis el programa en un fichero de salida p4ej1.res
./p4ej1<p4ej1.dat>p4ej1.res
```

Hacéis esto para todos los ficheros.

Para confeccionar la memoria utilizáis Emacs, u otro editor ASCII, pero no MS-Word o editores de RichText. Creáis un fichero llamado P4_nombre_apellido1_apellido2.txt, y con Emacs escribís vuestro nombre y grupo de prácticas (José Pérez García. Grupo BL1) en la primera línea e incluís en él todos los ficheros por orden correlativo: p4ej1.cpp, p4ej1.dat, p4ej1.res, p4ej2.cpp, Escribís cualquier comentario a final de los ficheros de resultados del ejercicio correspondiente. Separáis cada ejercicio con una línea 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/cristina con

```
cd ..
```

Hacéis un archivo comprimido de P4, con vuestro nombre, para que no se confunda con P4.tgz original.

```
tar -zcvf P4_cristina.tgz P4
```

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

```
tar -ztvf P4_cristina.tgz
```

Ponéis este fichero en vuestro espacio web de la Universitat y lo salváis en disco o memoria USB. Tomad al menos dos medidas de precaución, para evitar pérdida de datos. Las unidades de diskette y USB se montan haciendo doble click sobre los iconos correspondientes de la carpeta Equipo del escritorio.

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 `P4_nombre_apellido1_apellido2.txt` como respuesta a la Práctica 4 en el Aula Virtual de Cálculo Numérico.

8. Ejercicios a presentar como memoria

1. Escribid un programa que defina dos variables `a=10` y `b=20`, y una función `intercambia(&a, &b)` que las intercambie utilizando referencias, es decir, después de la llamada de la función, `a=20` y `b=10`. Escribid otra función, `intercambia2(pa, pb)` que intercambie `a` y `b` butilizando los punteros `pa` y `pb` de `a` y `b`, respectivamente.
2. Escribid un programa que lea la dimensión de dos vectores de la misma dimensión, y los cree utilizando el operador `new`. Los valores de los vectores se pueden leer desde la entrada estándar o desde fichero, a elección. Seguidamente, debéis calcular el producto escalar de ambos vectores e imprimirlo por la salida estándar (se puede utilizar una función para calcular el producto escalar o hacerlo directamente en `main()`). Finalmente, los vectores se deben destruir con `delete` antes de finalizar el programa.
3. Escribid un programa que lea tres `string` en la entrada estándar. Debe de crear un fichero con el nombre de la primera `string` y que contenga la segunda y la tercera `string` unidas. Debe de imprimir, además, la longitud de esta `string` total.

4. Sean los vectores $v1 = (1, 2, 3, 4)$ y $v2 = (-1, 0, 1, 0)$, y las matrices $A = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix}$

y $B = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$. Escribid un programa, utilizando la librería TNT, que calcule

$$C = A * A^T + B * B^T$$

$$C+ = B$$

$$v3 = 5 * v1 + (v1 * v2) * v2$$

$$v4 = A * v3 + B * v2$$

Definid cada variable del tipo que corresponda (ej. Vector v3). Imprimid los vectores y matrices resultantes en fichero.

5. Utilizando la librería `mpfun`, calcular un programa que calcule e imprima el factorial de 200 y el número de Fibonnaci 200 (de forma no recursiva).
6. Haced un programa que produzca una ventana con la gráfica de $\tan(x)$ entre -10 y 10 , utilizando la librería `gnuplot_i++`. La gráfica debe de permanecer 10 segundos en pantalla.