



ESCUELA SUPERIOR DE INGENIERÍA

INGENIERÍA INFORMÁTICA

ARQUITECTURA DE COMPUTADORES II

PRIMEROS PASOS CON CUDA

José Carlos Pavón Montañez
Pablo Pérez Luna

20 de mayo de 2010

Índice

1. Instalación	2
1.1. GNU/Linux	2
1.1.1. Drivers	2
1.1.2. CUDA Toolkit	2
1.1.3. SDK	2
1.1.4. Configuración	3
1.2. Windows	3
1.2.1. Drivers	3
1.2.2. CUDA Toolkit	3
1.2.3. SDK	3
1.2.4. Microsoft Visual Studio	4
1.2.5. Configuración	4
2. Primer programa	7
3. Compilación	8
3.1. GNU/Linux	8
3.2. Windows	8
4. Documentación	9
5. Preguntas frecuentes	10

1. Instalación

1.1. GNU/Linux

1.1.1. Drivers

Lo primero que tenemos que hacer es descargar los “Developer Drivers” de la siguiente dirección

http://developer.nvidia.com/object/cuda_3_0_downloads.html

Nos descargamos el los drivers que correspondan con nuestra versión de GNU/Linux. Para instalarlo tenemos que cambiar a una consola de sistema pulsando `ctrl+alt+f2`, por ejemplo. Una vez hecho esto nos logueamos y terminamos el entorno gráfico de la siguiente forma (como superusuario):

KDE `/etc/init.d/kdm stop`

GNOME `etc/init.d/gdm stop`

Ahora podemos proceder con la instalación del driver ejecutando como superusuario lo siguiente

```
sh NombreDelDriver.run
```

Lo único que nos queda por hacer es seguir los pasos del asistente.

1.1.2. CUDA Toolkit

Para instalar las herramientas de compilación ejecutamos lo siguiente como superusuario

```
sh NombreDelToolkit.run
```

1.1.3. SDK

También podemos instalar el SDK que contiene ejemplos y documentación sobre CUDA, pero esto es opcional.

1.1.4. Configuración

Para poder utilizar el compilador `nvcc` es necesario indicar donde se encuentra, para ello creamos el archivo `/etc/profile.d/cuda.sh` con el siguiente contenido

```
#!/bin/sh
export PATH=$PATH:/usr/local/cuda/bin
```

Solo nos queda indicar dónde se encuentran las bibliotecas para poder ejecutar los programas una vez compilados, para ello creamos el archivo `/etc/ld.so.conf.d/libcuda.conf` con el siguiente contenido

```
/usr/local/cuda/lib
```

1.2. Windows

1.2.1. Drivers

El primer paso que debemos seguir es descargar los últimos “Developer Drivers”. Los podemos encontrar en la página

http://developer.nvidia.com/object/cuda_3_0_downloads.html

Seleccionaremos Windows y el driver el adecuado según nuestra versión, diferenciando si es de 32 o 64 bits. Una vez descargados procedemos con la instalación siguiendo los pasos que nos proporciona el asistente.

1.2.2. CUDA Toolkit

Lo siguiente que tenemos que hacer es descargar e instalar “CUDA Toolkit” que contiene las herramientas necesarias (como el compilador `nvcc`) para compilar un programa de CUDA. Para ello procedemos de forma analoga a la sección anterior.

1.2.3. SDK

También podemos instalar el SDK que contiene ejemplos y documentación sobre CUDA, pero no es necesario.

1.2.4. Microsoft Visual Studio

CUDA en Windows utiliza el compilador `cl.exe` de Microsoft Visual Studio, por lo que deberemos instalarlo. Si no disponemos de ninguna licencia podemos obtener la versión Express que es gratuita y suficiente para nuestro propósito, aunque tendremos que registrarla, de forma gratuita también. Actualmente existen dos versiones Express, la 2005 y la 2010, pero la 2010 no está todavía soportada por CUDA, así que instalaremos la 2005. Para ello tenemos que entrar en la siguiente página web:

<http://msdn.microsoft.com/es-es/express/aa975050.aspx>

Buscamos Visual C++ 2005 Express Edition y seleccionamos nuestro idioma. Para instalarlo basta con seguir los pasos del asistente

1.2.5. Configuración

Lo que tenemos que hacer ahora es incluir el compilador `cl.exe` en el path del sistema, para ello seguiremos los siguientes pasos.

Primero haremos click con el botón derecho del ratón sobre Mi PC y luego en propiedades.

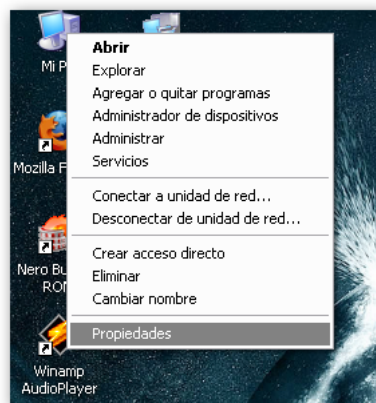


Figura 1: Paso 1

En segundo lugar seleccionaremos la pestaña “Opciones avanzadas” y haremos click en “Variables de entorno”.

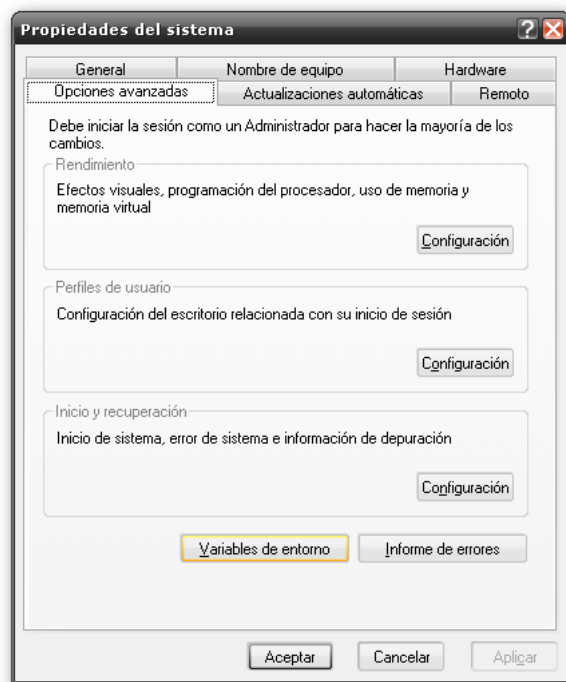


Figura 2: Paso 2

En la sección “Variables del sistema” buscaremos la variable “Path” y haremos click en “Modificar”.

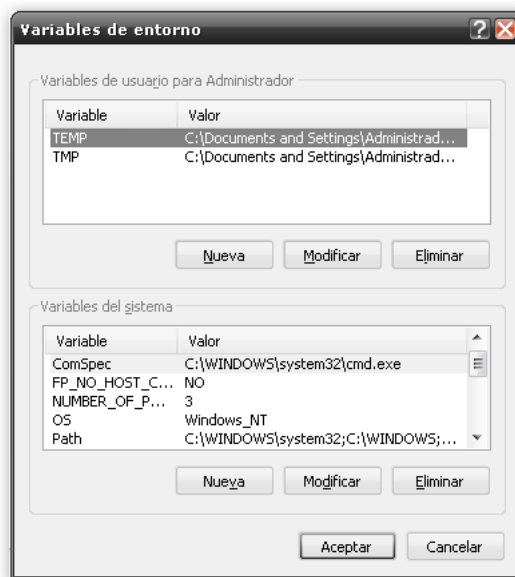


Figura 3: Paso 3

Añadiremos al final del campo “Valor de variable”, separado con punto y coma, la ruta completa del compilador del Visual Studio (`cl.exe`).

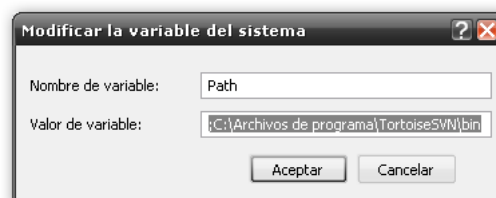


Figura 4: Paso 4

Una vez hecho todo esto el sistema estara preparado para compilar nuestro primer programa.

2. Primer programa

Para comprobar que la instalación se ha realizado correctamente, crearemos un nuevo programa que más tarde compilaremos. Para ello abriremos nuestro editor favorito y escribiremos el siguiente código. Al guardarlo es importante recordar que la extensión debe ser `.cu`.

```
1  #include <stdio.h>
2  #include <cuda.h>
3
4  main()
5  {
6      printf("Hola mundo\n");
7  }
```


3. Compilación

3.1. GNU/Linux

```
nvcc -o hola hola.cu
```

3.2. Windows

Si seguimos los pasos de la configuración del capítulo anterior bastará con escribir en la consola el siguiente comando:

```
nvcc -o hola.exe hola.cu
```

Si no pusimos la ruta del compilador en el path del sistema podemos indicarlo con la opción `-ccbin` de la siguiente forma:

```
nvcc -ccbin ``ruta_compilador\compilador.exe`` -o hola.exe hola.cu
```

4. Documentación

- Página oficial de CUDA en español:

`http://www.nvidia.es/object/what_is_cuda_new_es.html`

- Descargas de CUDA:

`http://developer.nvidia.com/object/cuda_3_0_downloads.html`

- GPUs con soporte para CUDA:

`http://www.nvidia.es/object/cuda_gpus_es.html`

5. Preguntas frecuentes

- ¿A qué es debido el error “nvcc fatal : nvcc cannot find a supported cl version. Only MSVC 8.0 and MSVC 9.0 are supported”?
Se debe a que se está utilizando el compilador de Microsoft Visual Studio 2010 que todavía no está soportado. La solución es instalar una versión anterior.
- ¿A qué es debido el error “nvcc fatal : Cannot find compiler 'cl.exe' in PATH”?
Se debe a que no se ha añadido el compilador al path. La solución es añadirlo de la forma descrita en la sección de configuración del capítulo de instalación o indicar la ruta con la opción -ccbin.