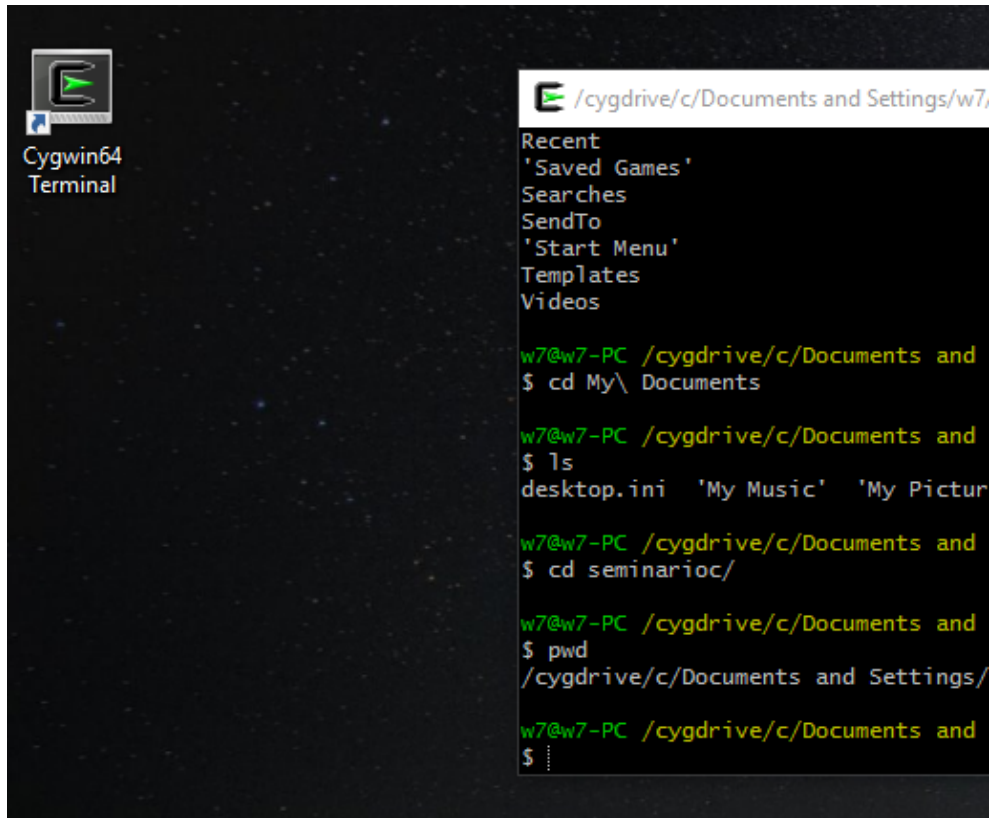


Introducción al uso de CLI (Command Line Interface)

Cygwin provee entre otras cosas, un intérprete de comandos al estilo Linux. Ese tipo de herramientas se conoce por muchos nombres: terminal, consola, línea de comandos, cmd, command, etc.



The image shows a Cygwin64 Terminal window. On the left, there is a file explorer view showing a directory structure. The main area of the terminal displays a series of commands and their outputs. The prompt is 'w7@w7-PC /cygdrive/c/Documents and Settings/w7/'. The commands and outputs are as follows:

```
w7@w7-PC /cygdrive/c/Documents and
$ cd My\ Documents

w7@w7-PC /cygdrive/c/Documents and
$ ls
desktop.ini 'My Music' 'My Pictur

w7@w7-PC /cygdrive/c/Documents and
$ cd seminarioc/

w7@w7-PC /cygdrive/c/Documents and
$ pwd
/cygdrive/c/Documents and Settings/

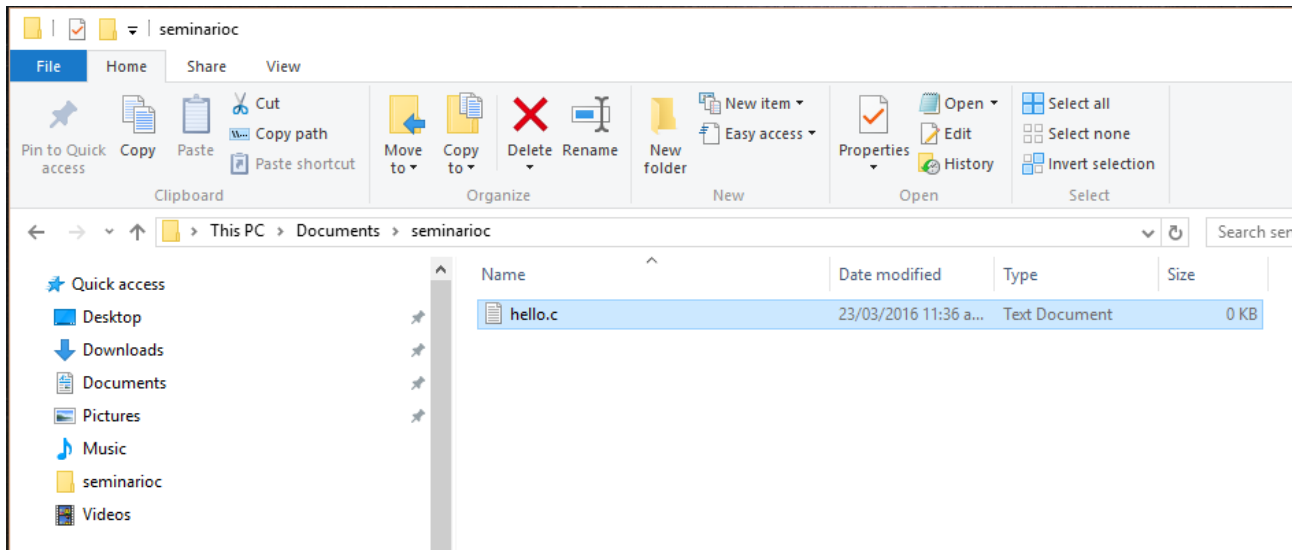
w7@w7-PC /cygdrive/c/Documents and
$
```

No debe sorprendernos que sea necesario usar la consola de Cygwin interactuar con las herramientas que instalamos con Cygwin (gcc, gbd, make, etc.). La consola nativa de Windows (conocida como cmd o Command Prompt) no reconoce algunos de los comandos y va a causar problemas.

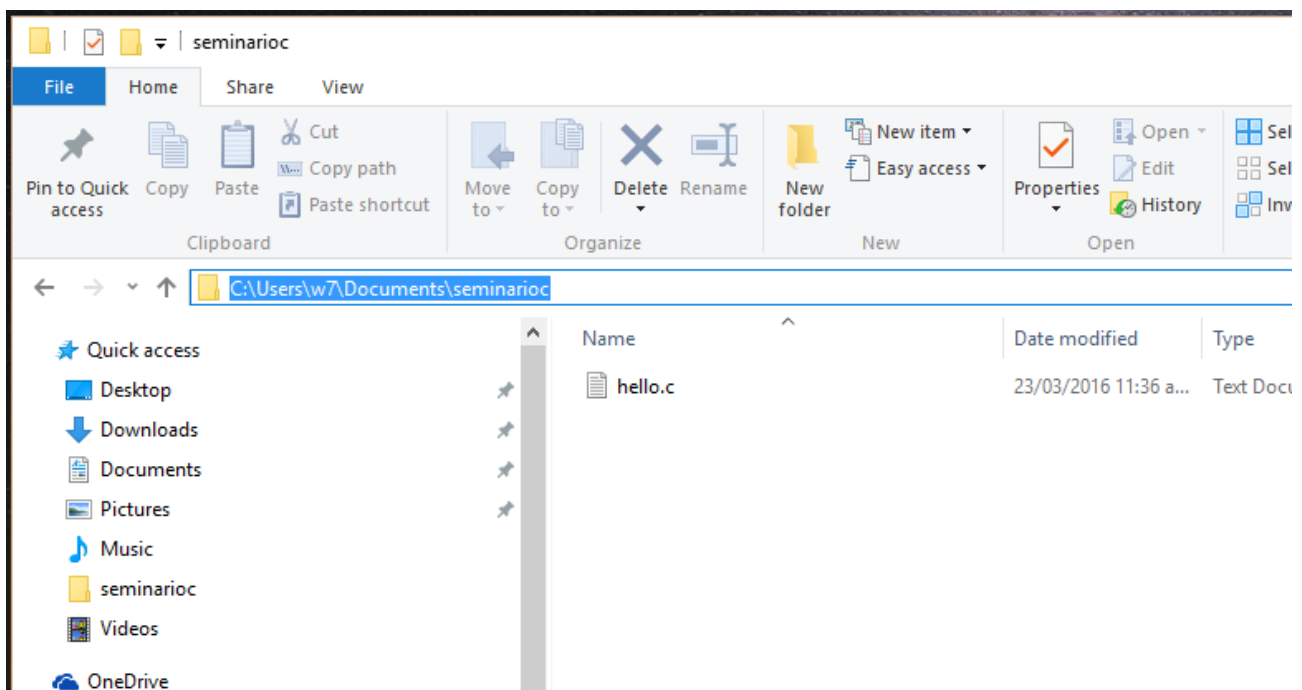
Para usar las herramientas de desarrollo de C en Cygwin tenemos que saber algunos (pocos) comandos y entender algunos conceptos básicos.

Lo primero solemos hacer al programar es crear algún archivo fuente, por ejemplo "hello.c" y lo guardamos en alguna carpeta como por ejemplo "seminarioc" dentro de "Mis Documentos".

Para acceder a esa carpeta desde Cygwin tenemos que averiguar el "path" (a veces conocido como "la ruta").



Apretando **CTRL+L**



Ya tenemos el path de la carpeta en Windows: "C:\Users\w7\Documents\seminarioc".

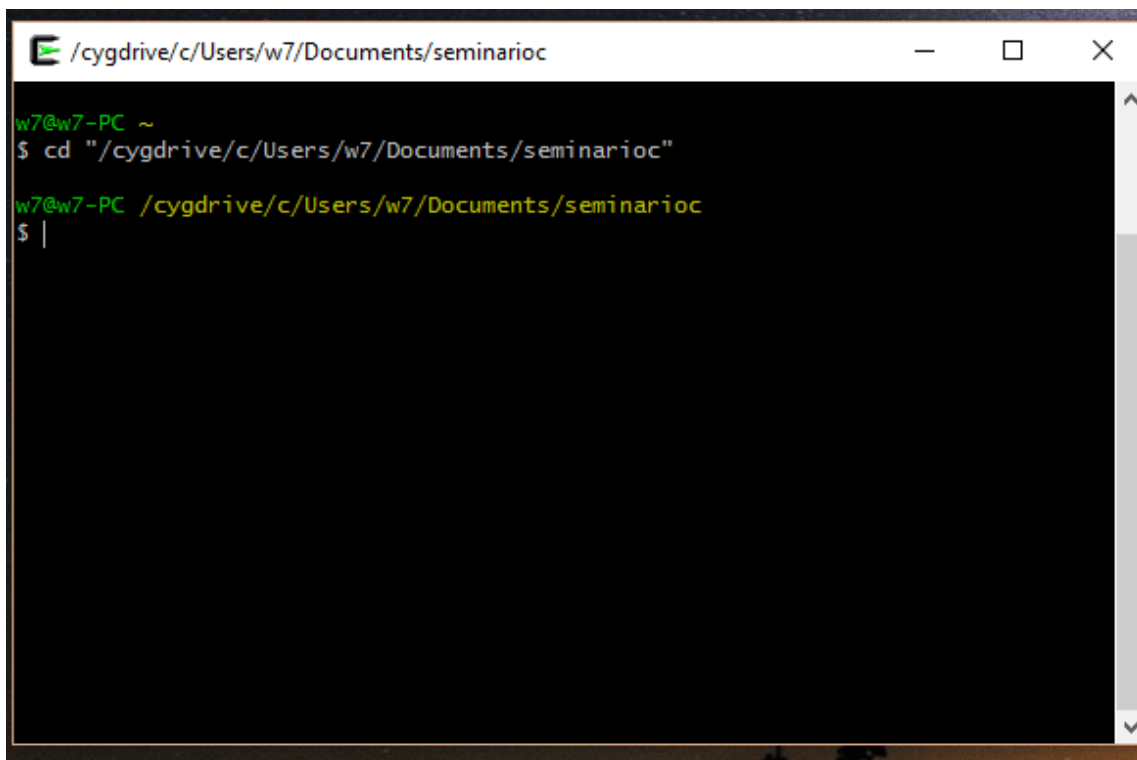
En Cygwin las carpetas se conocen como "directorios" y siempre existe un directorio seleccionado conocido como "directorio actual", "directorio de trabajo" o "el directorio en el que estoy parado".

Al abrir la consola Cygwin podemos usar el comando pwd (Print Working Directory) para saber en qué directorio estamos parados.

Para cambiar el directorio actual se usa el comando `cd` (Change Directory). Hay muchas formas de cambiar de directorio. Se le puede indicar el path completo de un directorio o un path relativo al directorio actual.

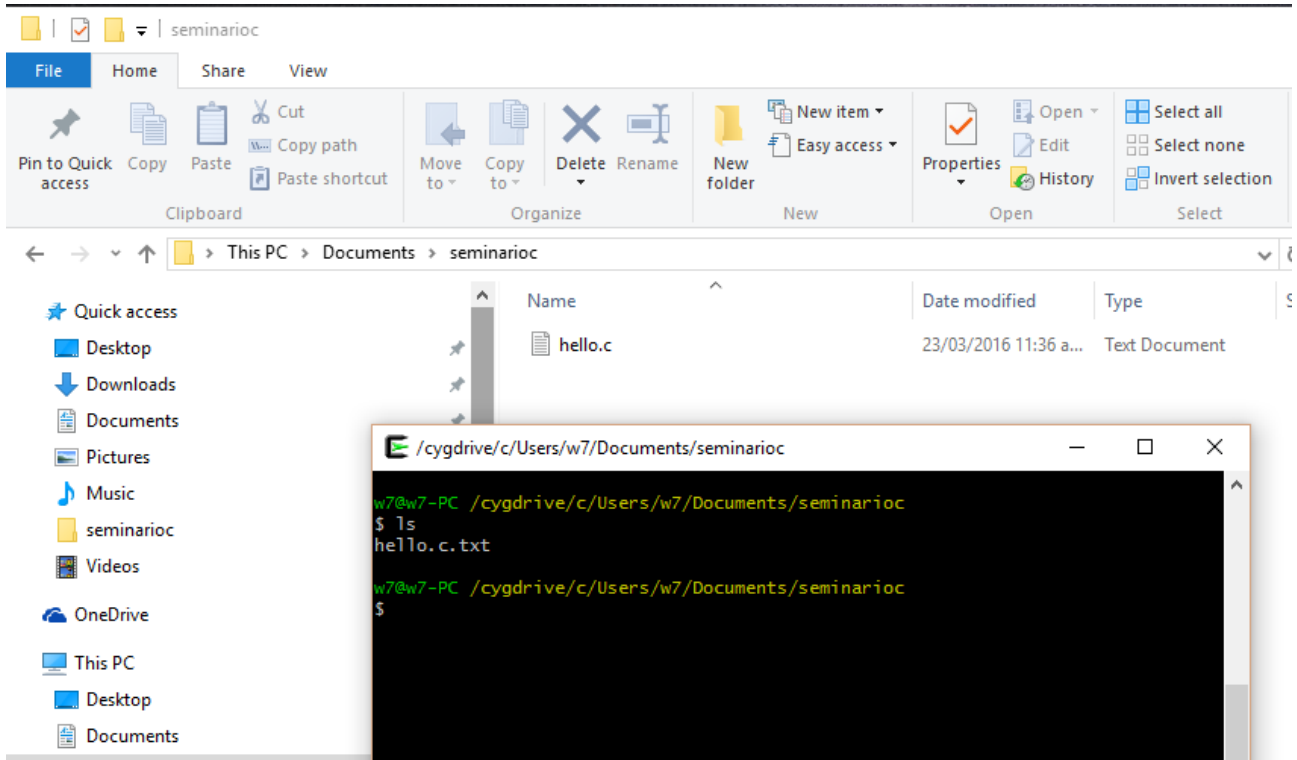
Para ir al directorio `"C:\Users\w7\Documents\seminarioc"` el comando es

`cd "/cygdrive/c/Users/w7/Documents/seminarioc"`

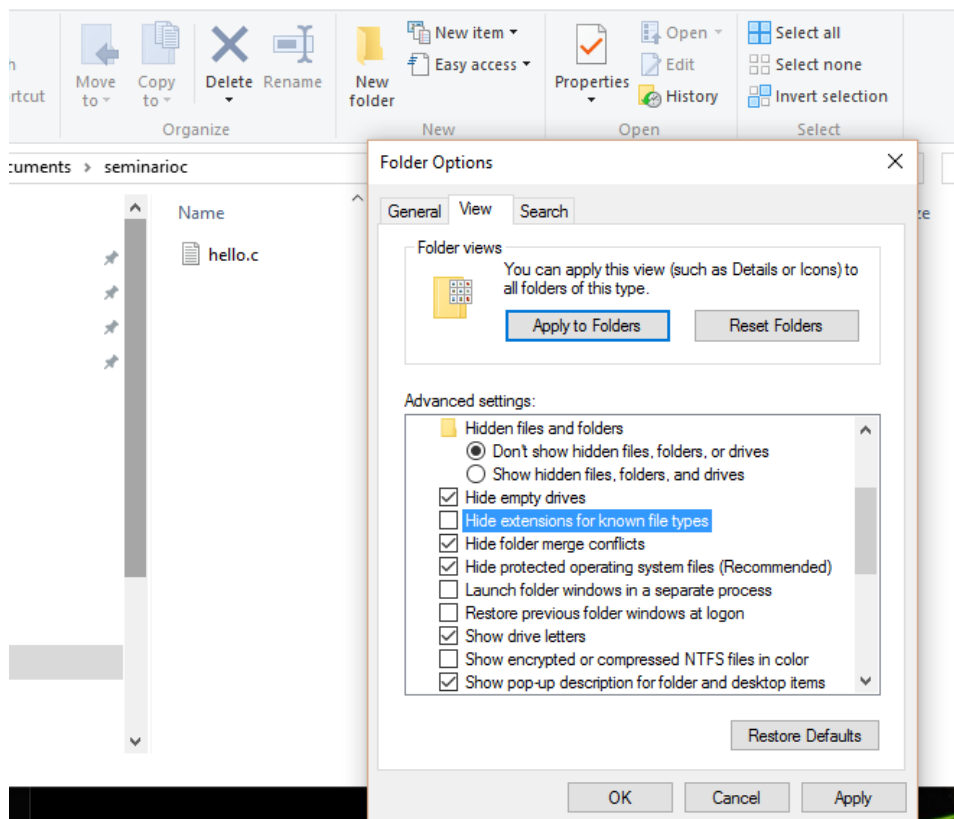
A screenshot of a Cygwin terminal window. The title bar shows the path `/cygdrive/c/Users/w7/Documents/seminarioc`. The terminal content shows the prompt `w7@w7-PC ~`, followed by the command `$ cd "/cygdrive/c/Users/w7/Documents/seminarioc"`. The next line shows the prompt `w7@w7-PC /cygdrive/c/Users/w7/Documents/seminarioc`, indicating the directory change was successful. A cursor is visible on the line `$ |`.

En este caso usamos el path completo también conocido como path absoluto (porque no es relativo al directorio actual). Las comillas son necesarias si hay algún espacio en el nombre de alguna carpeta, pero no están de más. Notemos que en Windows el separador de carpetas en el path es `"\"` pero en Cygwin es `"/"`.

El comando `"ls"` lista en pantalla los nombres de los archivos que están en el directorio actual. Es útil para verificar el nombre de los archivos. En el ejemplo se ve que lo que el explorador de archivos de Windows muestra como `"hello.c"` es en realidad `"hello.c.txt"`.



Es muy conveniente cambiar esta opción en Windows para ahorrarnos dolores de cabeza.



Una vez que el directorio actual en cygwin es el deseado podemos usar los comandos gcc, make, etc. indicando simplemente el nombre de los archivos fuente:

```
gcc -std=c11 -o hello hello.c -Wall
```

Consejos

1. La tecla “flecha hacia arriba” permite repetir comandos que se ejecutaron anteriormente.
2. La tecla “tab” autocompleta nombres de archivos a medio escribir.
3. En Windows los nombres de archivos no distinguen mayúsculas de minúsculas. Un archivo llamado Hello.c se puede referenciar en la línea de comandos como hello.c o hElLo.C. En Cygwin esto es igual porque se ejecuta en Windows, pero no es igual en Linux, así que es una muy mala práctica usar nombres de archivos con mayúsculas si se espera que esos fuentes sean compilados en Linux.