

Laboratorio 1

Jean Carlos Chavarría Hughes B11814

3 de septiembre de 2014

Resumen

Laboratorio 2 de el curso IE 0217.

1. Introducción

Este documento corresponde al reporte de segundo laboratorio del curso IE0217, en el cual se trabajó con distintas librerías de C++, tambien se practicó más con **makefiles** y además se trabajó con **doxygen**.

2. Comentarios Importantes

Librerías

La primera parte del laboratorio se enfocó en el uso de las librerías y como compilarlas y ejecutarlas. Se trabajó con el compilador **g++** y las banderas **-o** y **-c**.

1. Porqué se llama el programa de esta manera y no simplemente digitando hola.
2. Qué debe hacerse para poder correrlo escribiendo simplemente hola.
3. Porqué el devolver un 0 es signo de ejecución correcta y un valor distinto indica una condición de error.

Solución

1. Sucede porque el directorio actual no se encuentra en el **PATH**.
2. Se debería agregar al **PATH** del sistema, pero por seguridad no está, a diferencia de *Windows*. Una manera de hacerlo es con el comando:

```
PATH=.: $PATH: .
```

3. En esta pregunta, el devolver un cero funciona como símbolo de una correcta ejecución, meramente por convención, ya que en realidad el programa compila y se ejecuta de manera correcta, con sea lo que sea que tenga el *return*. Se hizo una prueba donde se eliminó el

```
return 0
```

y el programa funcionó correctamente.

Makefile

Se debe tener especial cuidado a la hora de copiar el texto de **makefile**, esto debido a que algunos caracteres se copian como si fueran otros y además que se incluyen espacios extras donde no deberían existir.

El problema que me surgió con el makefile fue que estaba ejecutándolo pero me presentaba errores, debido a que *el archivo o directorio <-r>no existía, también con el <-o>y el -c.*

Esto era debido a que al realizar copiar y pegar, el guión se copiaba como un guión largo, diferente al que se necesitaba y fue un error muy difícil de observar.

Doxygen

Tal como se explica en la guía del laboratorio, **Doxygen** es una herramienta generadora de documentación que soporta múltiples lenguajes de programación.

Algunos de los comandos más importantes para mencionar son:

```
doxywizard ejemplo.cpp
```

La línea anterior se encarga de generar el documento fuente para la compilación del doxygen, para lo cual se permite utilizar la interfaz gráfica. Sin embargo también es posible configurar sin interfaz gráfica. Este documento generalmente se gestiona con una extensión **.cnf**.

```
doxygen doc.cnf
```

En este caso, el fichero *doc.cnf* fue el nombre que se le dio al archivo de configuración. Y solamente falta generar los manuales, para lo cual se ejecuta el comando anterior.

Todos estos pasos fueron automatizados mediante el uso del archivo **makefile**,

como se puede observar en la carpeta *parte3*.