

INSTALACIÓN Y COMPILACIÓN DE VTK y FLTK USANDO DEBIAN/UBUNTU

Instalación de paquetes requeridos:

VTK es una librería para imágenes 3D, procesamiento de imágenes y visualización . Los paquetes a instalar son:

- **build-essential:** permite instalar todas las herramientas de desarrollo de Linux.
- **libvtk5:** librería para correr los programas que utilizan vtk.
- **libvtk5-dev:** librerías para incluir en nuestro programa y desarrollar con vtk.
- **libfltk1.1:** librería para correr los programas que utilizan fltk.
- **libfltk1.1-dev:** librerías para incluir en nuestro programa y usar fltk.
- **libfltk1.1-dbg (opcional):** permite realizar debuggin (paso a paso) a nuestros programas que utilicen fltk.
- **sun-java6-jdk (opcional):** Máquina virtual de java.
- **cmake:** Software para compilar programas multiplataforma y automatizar el enlace de librerías. (en otras palabras, para facilitarnos la vida buscando las librerías automáticamente).

Se pueden instalar cada uno de estos programas con el programa **Gestor de Paquetes Synaptic** (para hacerlo de manera gráfica) o utilizando el siguiente comando como root:

```
# apt-get install cmake libvtk5-dev build-essential libfltk1.1-dev libfltk1.1-dbg sun-java6-jdk
```

Compilar nuestros programas usando VTK:

Generando el Makefile:

Primero se necesita generar un Makefile (se recomienda buscar en internet para mas información) que automatizará el proceso de compilación y enlace de las librerías vtk, para ello se debe crear un directorio donde trabajaremos con el proyecto y en esa carpeta se debe crear un archivo con el nombre

CMakeLists.txt que contiene lo siguiente:

```
cmake_minimum_required(VERSION 2.6)
PROJECT( NOMBRE_PROYECTO )

FIND_PACKAGE ( VTK )
IF ( VTK_FOUND )
INCLUDE( ${USE_VTK_FILE} )
ENDIF( VTK_FOUND )

FIND_PACKAGE(FLTK)
IF(FLTK_FOUND)
  INCLUDE_DIRECTORIES(${FLTK_INCLUDE_DIR})
ENDIF(FLTK_FOUND)

SET(CMAKE_BUILD_TYPE "Debug")

INCLUDE_DIRECTORIES(
${CARPETA_CON_COD_FUENTE}
)

ADD_EXECUTABLE( NOMBRE_BINARIO COD_FUENTE.cpp)

TARGET_LINK_LIBRARIES ( NOMBRE_BINARIO
vtkRendering vtkGraphics vtkHybrid
vtkImaging vtkIO vtkFiltering vtkCommon
)
TARGET_LINK_LIBRARIES(NOMBRE_BINARIO ${FLTK_LIBRARIES})
TARGET_LINK_LIBRARIES(NOMBRE_BINARIO ${OPENGL_LIBRARIES})
```

Donde es importante reemplazar las siguientes palabras por lo necesario:

- **NOMBRE_PROYECTO** (Colocar el nombre del proyecto, que si lo hicieron con Netbeans debe usarse el mismo nombre respetando mayúsculas y minúsculas)
- **CARPETA_CON_COD_FUENTE** (Opcional, pero si tenemos los códigos fuentes dentro de

Descargado de: <http://skatox.com/>

otra carpeta, aquí se debe colocar el nombre de la misma)

- **NOMBRE_BINARIO** (Nombre del ejecutable que se va a generar)
- **MAIN.cpp** (Nombre del archivo cpp que contiene el main)

Una vez creado el archivo **CMakeLists.txt** nos vamos a esa carpeta por consola y ejecutamos el comando “**cmake .**” para generar el archivo **Makefile** que va a ser utilizado después:

```
$ cmake .
```

Compilar por consola

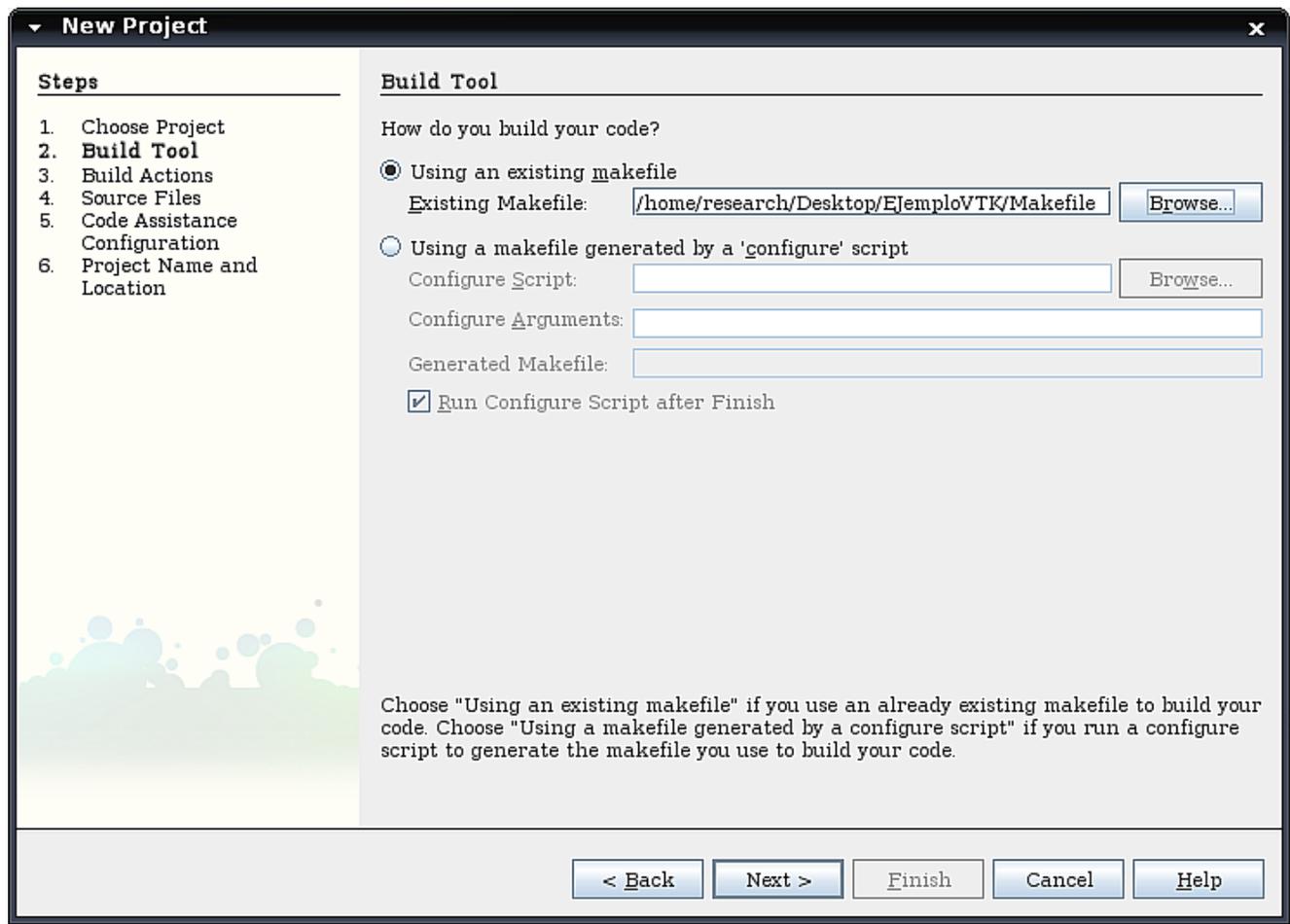
Simplemente se debe copiar el archivo **Makefile** generado anteriormente a la carpeta con el código fuente, al momento de compilar se debe ejecutar el comando **make** y luego correr el programa con el nombre que definimos en el **CMakeLists.txt** :

```
$ make  
$ ./NOMBRE_BINARIO
```

Por lo que para generar el binario llamado Programa con el archivo Main.cpp basta con ejecutar la primera línea para compilar y la segunda para correr el programa.

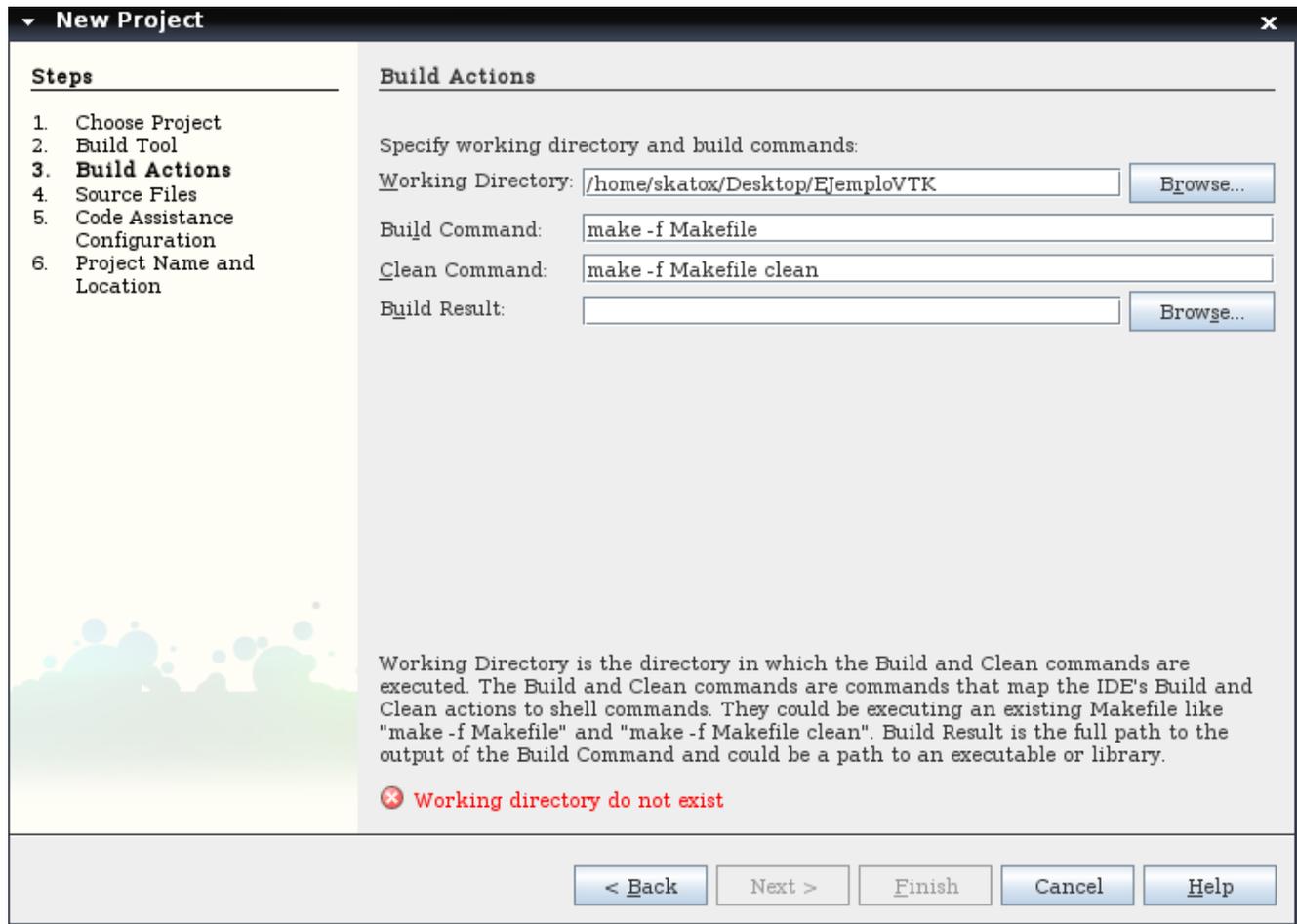
Compilar usando Netbeans

Aquí ya se debe tener generado el Makefile, luego al abrir Netbeans tenemos es que crear un Proyecto de C++ con fuentes existentes (C/C++ Project from existing Code), luego presionamos siguiente y donde dice "Using an existing Makefile" buscamos el archivo con el Makefile generado con cmake tal como se muestra a continuación:



Descargado de: <http://skatox.com/>

Luego tenemos que definir el directorio o carpeta donde vamos a desarrollar, simplemente navegamos al lugar de nuestra carpeta (si no existe se crea una) le damos a siguiente y listo:



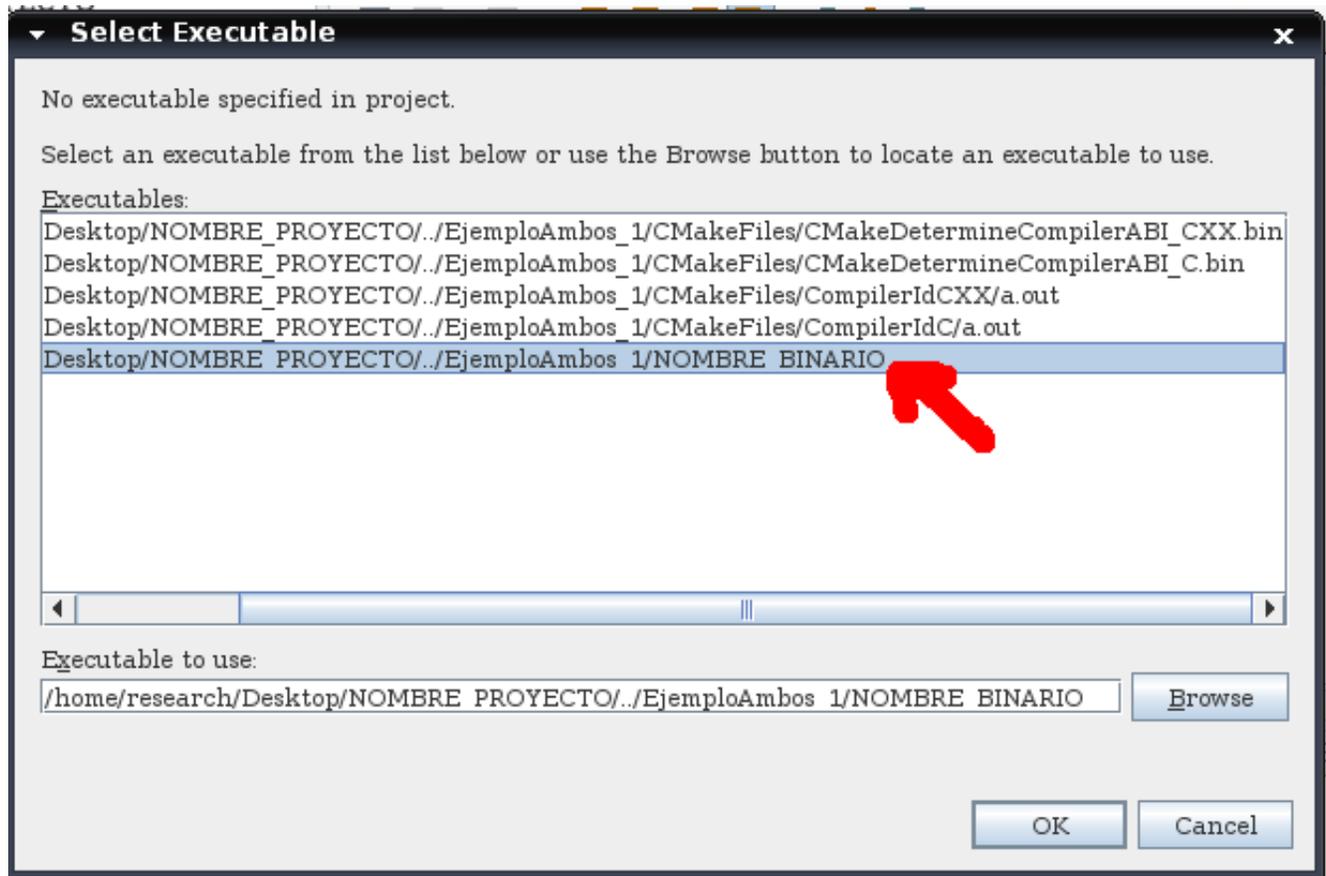
Luego el proceso es igual para cualquier proyecto de C++, especificar la carpeta con las fuentes del proyecto, pasarle argumentos al compilador, definir el archivo main y las rutas de trabajo (en otras palabras, siguiente todo el tiempo).

NOTA: Al compilar por primera vez puede ser que pregunte por el archivo ejecutable, simplemente seleccionamos aquel que definimos en el

Descargado de: <http://skatox.com/>

Descargado de: <http://skatox.com/>

CMakeList.txt (ver siguiente página para ejemplo)



**Seleccionamos el binario que correspondiente al definido
previamente con cmake**

Descargado de: <http://skatox.com/>