**Tutorial 14**

Ya es necesario empezar a desarrollar nuestro código del kernel usando un lenguaje de más alto nivel que solo ensamblador. Para nuestro caso será utilizar el IDE de Microsoft Visual Studio C++ 2005-2010. Para mi proyecto he conseguido el Visual C++ 2010 y a parte de las recomendaciones de compilador y de linkeador, hay otros parámetros que tocar como el Output Directory, quitar el DEP y Dynamic Base y agregar una variable mas aparte de “/ALIGN 512”al final de la configuración del linker.

En C/C++ se utilizan mucho las librerías para poder referenciar código de clases y poder utilizar este código de clases especificas en otras clases u objetos. El proyecto desarrolla un cierto tipo de “Caja de Pandora” como lo menciona en el tutorial y hace referencia a poder utilizar las librerías y proveer interfaces externas a otro código de C para así mantener “encapsulado” el código principal de cada una de las clases.

Una vez definido como trabajaremos es importante hacer una librería estándar para nuestro sistema operativo. Ya que no podemos usar librerías de C/C++ ampliamente conocidas porque nuestro sistema operativo simplemente no aporta las llamadas al sistema adecuadas para hacerlas funcionar, tenemos que crear nuestroas propias librerías. Primero definimos los tipos de datos que vamos a utilizar en nuestro sistema operativo como por ejemplo null, size\_t, cuando mide un signed char, unsigned char y asi consecutivamente con los otros tipos de datos básicos. Esto es todo por el tutorial 14, no se cubre mucha teoría de sistemas operativos como tal, sino que trata de preparar el ambiente de desarrollo y ya con el siguiente tutorial poder realizar algo de teoría y ponerlo a la práctica.