

Bastante he leído de C++, y he visto todo tipo de verdades, mitos y leyendas de este magnífico lenguaje. Francamente, salvo una poca literatura que es realmente excelente, la mayoría de lo que he leído son puros cuentos. Y en este pequeño artículo presentaré al lector algunas muestras de ello, y mostraré la verdad para tratar de erradicar tanta cosa mala que se dice.

Muy breve historia de C++.

Fue creado en 1983, en los laboratorios Bell (en aquellos años parte de ATT) por un programador llamado Bjarne Stroustrup que necesitaba un lenguaje de más alto nivel que C pero aun lo bastante compatible con C para no tirar a la basura el código ya escrito en C. Nunca existió un proyecto C++, nunca pretendió ser orientado a objetos ni ser un lenguaje de programación genérica, mucho menos ser el lenguaje más extendido y por supuesto que no buscaba competir con C. Su humilde origen fue hacer la vida más fácil a un entusiasta programador, y que tuvo el atino de hacerlo libre y compartirlo con todos los demás. No hay una gran empresa detrás de él y el comité que lo norma no tiene fines de lucro.

Los mitos acerca de las versiones y su estandarización.

El origen.

Históricamente, dado que C++ fue creado por un desarrollador de ATT fue natural que esta empresa diera mucho apoyo a su difusión, y así ha sido hasta nuestros días. Algo similar ocurrió con el lenguaje C y otros más (por ejemplo, SmallTalk que fue creado por Xerox). Pero un lenguaje de programación de amplia difusión, y libre, como lo es C++ no puede estar sujeto a los caprichos de una empresa, sino normado por un comité o grupo que canalice las exigencias y necesidades de la comunidad. Así que, era de esperarse que comenzara (más tarde que temprano) su estandarización.

Las dificultades en su estandarización.

La estandarización comenzó en 1991, y se dio paso a la difícil tarea de escuchar a los programadores que conformaban la comunidad de C++ y contruir las bibliotecas del lenguaje, afinar ambigüedades, etc. Pero desgraciadamente el proceso fue muuuuuuuuuuu lento. Así, ocurrió que la versión "estándar" de C++ fue por mucho tiempo la que se incluía con los compiladores de ATT. El proceso de estandarización terminó en 1998 y se tuvo así el lenguaje Ansi C++ (normado para ser usado en Norteamérica) y el ISO C++ (para uso internacional). ¿Y el panorama hoy? Francamente desalentador, la mayoría de los supuestos (o que se hacen llamar) programadores de C++, en realidad lo que conocen es el ATT C++, el cual fue muy popular y difundido gracias a Borland y su magnífico (en aquellos días) Turbo C++ (¿acaso existe alguien que no lo haya usado una vez?). Por otro lado, la mayoría de los compiladores de hoy en día son deficientes en sus implementaciones del Ansi, tanto a nivel lenguaje como al nivel de las bibliotecas (la famosa STL es un ejemplo de las bibliotecas de C++ más poderosas y versátiles).

C++ y otros cuentos

Escrito por adrianvaca

Domingo, 20 de Marzo de 2011 12:51 - Actualizado Domingo, 20 de Marzo de 2011 12:53

¿Y en que nos afecta? Pues que por malas características del ATT C++, se creó una mala fama y muchos huyeron como si conocer C++ fuera diabólico. Y en el otro lado, muchas aplicaciones (como Mozilla) están escritas en un C++ antiguo y que carece de la riqueza que hoy tenemos en el Ansi C++.

□ Nunca en mi humildad, merecí tanto honor ni tanta ingratitud □.

Ésta es la adaptación de una frase con la que Bertrand Meyer, expresa lo que es C++. Y es que C++ ha cumplido con su difícil misión de servir de transición entre la programación estructurada y las actuales técnicas de programación, sin tener nunca la aspiración pomposa de lenguajes patrocinados como Java (por SUN) ó C# (por Microsoft).

Pero no seamos pesimistas, tenemos a gcc y aunque no es perfecto, compila muy bien casi todo y tiene pocos fallos en su biblioteca, y el panorama a corto plazo es excelente, ya que la salida de gcc 3 (al momento de escribir ésto, la versión oficial es la 2.95) es inminente y está cargado de muy buenas nuevas, ya que por fin cumplirá no sólo con el estándar de C++, sino también con el nuevo C99 (la versión más nueva del lenguaje C) pero C99 será objeto de un futuro artículo.

El mito de la curva de aprendizaje.

Tradicionalmente se dice que C++ es muy difícil de aprender. Dependiendo del punto de vista que se tome, ésto puede ser cierto o falso. Veamos por qué. Hay dos opiniones generalizadas y encontradas: en una se dice que un lenguaje debe tener los elementos de modo que un programador se adapte y lo domine por completo (y es el diseño que tiene lenguajes como Python), y otros afirman que un lenguaje debe proporcionar los suficientes elementos, para que un programador adopte sólo un pequeño subconjunto del lenguaje, a los cual se le conoce como dialecto de un lenguaje, y la comunidad de C++ es de esta opinión, y por ello tiene tantos elementos y recursos para el programador, pero ello implica que pocos llegan a conocer todo el lenguaje, y sin embargo la mayoría se acostumbra a usar algún dialecto (por ejemplo, el ATT C++ es un dialecto de C++, y de hecho es el dialecto o subconjunto más usado y conocido).

Hay varios dialectos en C++, y destacan el dialecto orientado a objetos, el dialecto de programación genérica, y el orientado a cálculo matemático.

Entonces, C++ realmente no es difícil de aprender, siempre que aceptemos que es estúpido querer conocerlo todo de manera abstracta, y que es mejor aprender en qué casos se usa cada dialecto. Por ejemplo, Java se puede hacer corresponder con el dialecto orientado a objetos, y Fortran se puede hacer corresponder con el dialecto orientado a cálculo numérico. Cada dialecto tiene sus ventajas y desventajas, y los dialectos nunca usan todo el lenguaje, por ello son generalmente muy fáciles de aprender y cada uno tiene sus propios defensores y detractores.

El mito de la eficiencia.

Es innegable que C++ compila código muy eficiente y pequeño en tiempo de ejecución, pero no

C++ y otros cuentos

Escrito por adrianvaca

Domingo, 20 de Marzo de 2011 12:51 - Actualizado Domingo, 20 de Marzo de 2011 12:53

sólo debemos ver la rapidez porque a menudo nos importará la rapidez de desarrollo. Aquí C++ cumple un papel fundamental, porque se optimiza casi tan eficientemente como C, pero con la ventaja de ser mucho más fácil de mantener y con menos esfuerzo. Por sus características de poseer una biblioteca de algoritmos, contenedores, manejo automático de memoria, orientación a objetos, genericidad, acceso a recursos de bajo nivel, y tantas más, es que escribir programas en C++ es generalmente menos penoso que en C.

Un caso concreto es la biblioteca de programación GTK--. Ésta está escrita en C++ y subyace en la librerías de GTK en C. Sabemos que GTK es orientada a eventos, así que a cada evento le asociamos una cierta función de respuesta. Pero en C, esas funciones se manejan como punteros arbitrarios y por lo tanto si nos equivocamos al indicar una función de respuesta, será hasta al momento de ejecutar el programa cuando nos ponga una violación de segmento o alguna gracia similar; en cambio, en C++ esas asociación se revisa en tiempo de compilación y así explícitamente podemos ver el error y donde fue originado, y se nos sugiere donde corregirlo.

Otro ejemplo, y el más severo, es que C++ nos ofrece un manejo de memoria que se nos presenta en varias categorías que podemos elegir, desde al más rudimentario estilo de C hasta al alto nivel de Java con su recolección automática de memoria. Y es que que C es muy propenso a errores de fugas de memoria, que son difícil de encontrar, en cambio en C++ todavía nos damos el lujo de elegir. Aquí se desmiente un mito: que la memoria es también un problema en C++, cosa falsa si quienes dicen éso se documentan.

El mito de la propiedad

En mi opinión, éste es el peor mito. Hay quienes afirman que C++ es propiedad de Microsoft, y que fuera de Visual C++ (el cual es malísimo y tiene de los peores compiladores) no hay cosa más. Por ello, tienen la asociación: Visual C++ = C++. Ello es absurdo. C++ es libre y no hay que pagar licencias por usarlo, C++ tampoco en un paquete que se pueda descargar ¿O acaso pueden bajar el lenguaje Español de internet? Claro que no, similarmente, C++ por ser un lenguaje no se puede descargar. Lo que descargamos son los compiladores y sus bibliotecas.

Esta confusión surge por la presión monopólica de Microsoft en el mercado, y por el efecto Java causado porque SUN es dueño de Java y determina su destino, así que es comprensible que el novato se confunda, pero para aclarar éso estamos nosotros.

Conclusión

Hemos visto que C++ ha evolucionado mucho desde su salida en 1983 cuando se le llamaba simplemente `□ C con Clases □` (fue hasta 1985 que a alguien se le ocurrió ponerle C++) y desde aquellos años a la fecha aun existen muchos mitos y falacias acerca de ello. He analizado algunos de esos mitos y he tratado de explicar por qué no son del todo verdad en nuestros días. Se ha vislumbrado la salida de gcc 3 como la más importante versión de gcc por los beneficios que traerá.

C++ y otros cuentos

Escrito por adrianvaca

Domingo, 20 de Marzo de 2011 12:51 - Actualizado Domingo, 20 de Marzo de 2011 12:53
