Tearing apart printf()

BY:
JOSÉ CARLOS ALCANTARA HOYOS
&
ROEL ALEJANDRO PEREZ CANDANOZA

El lenguaje de programación C puede llegar a ser muy complejo y robusto para realizar tareas gracias a su compilador.

Una de las primeras funciones utilizadas por la mayoria de los programadores principiantes se trata de la función printf() la cual se encarga de imprimir a consola el texto que se introdujo como parte de los argumentos de la función.

El articulo *Tearing apart printf()* nos ayuda a comprender paso a paso cómo es que se lleva a cabo paso a paso la impresión a pantalla, desde la escritura del código en C, hasta llegar al resultado en consola.

En el siguiente cuadro sinóptico se describen los puntos más importantes del artículo.

Código fuente

Incluir biblioteca stdio.h
Esta biblioteca incluye la función printf(), e incluirla es el primer paso para su implementación en código,

Usar printf()
Para asegurarnos que el compilador use esta funcion, debemos escribir una cadena que contega algun formato, de lo contrario, si tenemos una cadena simple, el compilador usara otra función de manera interna.

Compilando...

Código objeto
En esta parte, el compilador genera el código del programa usando la extensión .o , pero no sabe aún cómo definir printf, por lo que lo deja para ser resuelto en pasos posteriores

Ligador
El ligador procede a resolver el problema generado en la construcción del cod. objeto, por lo que une el código objeto de nuestro programa con el de la biblioteca.
Si se hace de forma estática, se importaría toda la biblioteca, mientras que si se hace de manera dinámica se liga principalmente lo que solo incluya el runtime de C, lo que falte se resuelve con el cargador.

Cargador
El cargador realiza lllamadas al sistema, las cuales son utiles para cargar las bibliotecas necesarias para ejecutarse y procede a cargar el programa a memoria.
Si se ligó el programa de forma dinámica, el cargador tendra que hacer muchas llamadas al sistema, y cargará las bibliotecas necesarias a memoria. Por otro lado, la forma estática realizará menos llamadas al sistema y solo se cargara ese programa a memoria, esto debido a que ya contiene todo en un mismo programa.

Ejecutando...

Parseo
Se procede a leer la cadena, se inicia de izquierda a derecha, y con una funcion especial para tratar al buffer llamada vfprintf(), se analiza la cadena, si se trata de un caracter, algun formato especificado o el caracter de terminación. Una vez ha sido analizada la cadena, se guarda en el buffer de salida.

Standard Output
Con la cadena parseada en el buffer como parte de stdout., está lista para ser pasada a consola, generalmente línea por línea. stdout conduce a la llamada al sistema "write". A pesar de que en el código fuente glibc hay muchas formas de escribir write, a nivel binario saltan al mismo código ejecutable. Write revisa el estado del hilo y, si todo está bien, mueve el número de write syscall (1).

Kernel
Una vez hecho lo anterior, pasa al kernel del SO para ser ejecutado por los drivers. El kernel se encarga de mandar a llamar operaciones que hacen que el proceso de la terminal genere la cadena en la consola, esto se hace mediante el cambio de una serie de banderas que le indican al buffer empujar nueva información o si puede añadir una nueva cadena en la ventana de salida.

Resultado

Cadena en consola...