

Ejercicio 1 - Estructuras de Datos y Algoritmos

26 de agosto de 2012

El ejercicio siguiente constará de 6 pasos. Idealmente le debería tomar unos 15 minutos realizar cada uno. Los contenidos a tratar serán los siguientes:

- TIPOS: char, int, float, strings, booleanos, overflow, arreglos.
- OPERADORES: diferenciar “=” y “==”, aritméticos, operador módulo, incrementar/decrementar, precedencia.
- ~~Creación de MAKEFILES.~~
- Búsqueda y utilización de las funciones de *gnu libc*. (Uso de documentación)
- CONTROL DE FLUJO: condicionales e iteraciones.
- FUNCIONES, empleando sólo paso por valor.
- Lectura y escritura de VALORES EN CONSOLA (funciones printf y scanf)
- USO DE LA TERMINAL, para compilar, ejecutar y depurar.
- Directivas de preprocesador. Uso de #INCLUDE.

Consideraciones previas al trabajo

- El código deberá estar escrito según el estándar de codificación GNU.
- El código escrito debe estar perfectamente indentado con tabuladores, no espacios.
- Algunos archivos de documentación no fueron cargados a la máquina virtual. En esta guía hay enlaces a esos documentos faltantes. Si los necesita, descárguelos y déjelos en el escritorio de la máquina virtual.

Recomendaciones

- Gran parte del código de esta tutoría se ejecutará dentro de un bloque WHILE. Trate de dividir su código en funciones de forma que este sea fácil de leer. Todo lo que necesita saber sobre funciones se encuentra en el punto 5 del manual de C que se encuentra disponible en el escritorio de su máquina virtual (*gnu-c-manual.pdf*.)

Guía

1. Escriba la estructura básica de un programa en C. Si no la conoce lea el punto 5.7 de *gnu-c-manual.pdf*. Luego escriba un mensaje de bienvenida utilizando la función `PRINTF` (*libc.pdf* pag. 253 + <http://goo.gl/0xrGh> 2.1). Finalmente compile el programa que acaba de escribir con GCC y ejecutelo utilizando DDD. Tenga en cuenta que para poder ejecutar el programa con DDD es necesario compilar el programa con información de debugging. Si no entiende algo de la última instrucción consulte los manuales del sistema con el comando `MAN`. Si aún tiene dudas o no sabe usar el terminal de Linux consulte al ayudante.
2. Se desea simular el funcionamiento de un ascensor. Defina un arreglo global (*gnu-c-manual.pdf* 2.5) de tipo `CHAR` que represente la cantidad de pasajeros esperando al ascensor en cada piso del edificio. El largo del arreglo representará la cantidad de pisos del edificio, codifíquelo de modo que se pueda adaptar el código a una cantidad arbitraria modificando sólo una constante (<http://goo.gl/0xrGh> 3.1). Luego modifique su función `MAIN` para que después de imprimir el mensaje de bienvenida se entre en un bucle que incremente aleatoriamente la cantidad de pasajeros cada 5 segundos. Ayúdese de *libc.pdf* para encontrar funciones que le permitan lograr esto. Además le servirá utilizar la sentencia `while` explicada en el punto 4.5 de *gnu-c-manual.pdf*. ¿que pasa si hay 1000 pasajeros en un piso?
3. Defina dos variables globales que representen la cantidad de pasajeros que están dentro del ascensor y el piso en que el ascensor se encuentra. Ambas variables deben comenzar con valor cero. Escriba un algoritmo para que el ascensor pase a buscar a los pasajeros a los pisos correspondientes. El ascensor debe poder moverse solo un piso cada 5 segundos! Además el ascensor soporta un máximo de 8 pasajeros.
4. Cree un arreglo para almacenar los pisos que han sido marcados en el ascensor. ¿De qué tipo debe ser el arreglo? Luego escriba un algoritmo que simule la elección del piso hecha por los pasajeros al momento de subir al ascensor. Modifique el algoritmo del punto 3 para que los pasajeros sean dejados en el piso que corresponde. Asegure que ningún pasajero se mantenga indefinidamente dentro del ascensor.
5. Escriba una función que en cada iteración del bloque `WHILE` (cada 5 segundos) muestre por pantalla el piso actual en que se encuentra el ascensor, la cantidad de personas que hay en el ascensor, el peso total de las personas que hay en el ascensor, la cantidad de personas que hay esperando en cada piso y los pisos que estan marcados en la botonera del ascensor. ¡Ojo, las personas pesan 73,327542 Kilos cada una!
6. Comente su código, escriba un `MAKEFILE` (<http://goo.gl/F8qCi>) y un *readme.txt*. Estos tres elementos facilitan la lectura, compilación y comprensión del código que está escribiendo. En el archivo *readme.txt* debe especificarse que es cada archivo de su programa, como debe ser compilado y como debe ser ejecutado. Además debe explicar cualquier particularidad que pueda tener su programa.

Nota! Al finalizar la tutoría pida a su ayudante que revise su trabajo. Luego limpie los archivos que haya creado en su máquina virtual. Asegúrese solucionar todas sus dudas antes de irse!