

Desarrollo de una terminal de comandos en lenguaje C

Una terminal de comandos es una aplicación que permite a un usuario enviar comandos, mediante alguna interfaz de comunicación, a una computadora o sistema embebido para realizar tareas de configuración o control. Es un mecanismo usual para interactuar mediante un puerto serie con sistemas que no poseen una interfaz gráfica de usuario por ejemplo.

Tarea:

Implementar en lenguaje C una ampliación de terminal de comandos que permita realizar las siguientes acciones:

- La aplicación va a manipular en memoria RAM un registro de 64 variables de 16 bits de tipo entero con signo cuyos valores serán asignados y leídos mediante comandos - La interfaz a utilizar será la entrada y salida estándar de la PC (STDIO) - Al inicio la terminal deberá mostrar el siguiente mensaje:
 - o `*** Shell Application vMM.mm.bb ***\n`
Donde MM.mm.bb indican el numero de la versión de la aplicación. Al inicio podrá ser 01.00.00 por ejemplo
(referencia: https://es.wikipedia.org/wiki/Versionado_de_software)
- Tratamiento de los comandos:
 - o Se usarán caracteres con codificación ASCII de 8bits
 - o Cada comando se procesará de a uno a la vez
 - o Todos los comandos deben devolver una respuesta
 - o Todos los comandos deben terminar con el carácter de fin de línea '\n' o Si un comando se ejecuta de manera exitosa pero no tiene respuesta, debe retornar `"OK\n"`
 - o Si hay algún fallo al ejecutar el comando debe retornar `"FAIL\n"`
 - o Todos los comandos deberán tener un máximo de 40 caracteres de longitud, incluidos los argumentos o parámetros de estos si existieren
- Formato de los comandos:
 - o `"Cmd arg1 arg2 ... argN \n"`
 - o Cmd: identificador del comando, secuencia alfanumérica única de máximo 10 caracteres
 - o argN: secuencia alfanumérica que indica el contenido o valor de cada uno de los argumentos o parámetros de los comandos. Puede contener comillas dobles si el argumento incluye espacios
 - o espacio: el carácter espacio en blanco es un delimitador, indica la separación entre el comando y cada uno de sus argumentos

Listado de comandos soportados:

Comando	Argumentos	Respuesta	Descripción
versión	Ninguno	MM.mm.bb	Devuelve el número de versión de la aplicación

echo	arg1 "cadena de texto"	arg1	Devuelve el argumento que puede contener espacios en cuyo
------	------------------------	------	---

			caso será enviado entre comillas dobles
set	arg1 = "Vn", n = [1..64] arg2 = "m", m = entero con signo	"m" o "FAIL"	Asigna el valor m a la variable n del registro en RAM
get	arg1 = "Vn", n = [1..64]	"m" o "FAIL"	Devuelve el valor de la variable n
reset	ninguno	"OK"	Pone todas las variables a cero

Se pide:

- utilizar lenguaje C de manera clara, adecuadamente tabulado
- utilizar funciones y variables con nombres cortos y significativos y el adecuado tipo de datos
- evitar el uso de variables globales e intentar mantener un consumo reducido de memoria para las variables
- documentar todos los archivos, funciones, variables y bloques de código de interés, se puede utilizar la herramienta Doxygen y su formato de documentación (<https://www.doxygen.nl/index.html>
http://www.disca.upv.es/aperles/mola_doxygen/mola_doxygen.html)