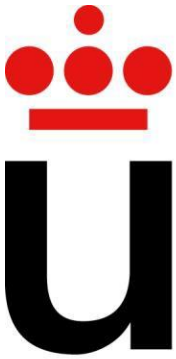

Universidad
Rey Juan Carlos

Sistemas Operativos

[PRÁCTICA 1 – PROGRAMACIÓN C] [23-24]

GUILLERMO BLÁZQUEZ BARBACID

TABLA DE CONTENIDO

Autores

Guillermo Blázquez Barbacid.

Descripción del Código

El código se compone de 4 archivos o librerías, La primera librería Llamada librería.C, se compone de las 3 funciones principales llamadas Head, Tail y Longlines. Siguiente archivo se llama librería.H la cual tiene todas las cabeceras de las funciones anteriores. Test.c que tiene la comprobación de argumentos y de errores, y el main del programa. Y un script llamado compila.SH el cual compila test y librería C, y ejecuta el programa.

Diseño del Código

Hemos declarado una lista simple, hemos creado una función de insertar final. Y un método de ordenación.

Mostrar es una función que muestra por pantalla para hacerlo más eficiente.

Head se encarga de leer todas las líneas de la entrada estándar y mostrar las N primeras. Creamos un bucle que todo lo que introduce por el buffer lo muestra, como la función head básica de C.

Tail se encarga de leer todas las líneas de la entrada estándar y solo mostrar las N últimas. Esta función guarda en una lista todas las líneas introducidas por el buffer en la lista con insertar final lo muestra las últimas.

Longlines se encarga de leer todas las líneas de la entrada estándar y mostrar las N líneas más largas de la más larga a la más corta. Esta función guarda una lista de todas las líneas introducidas por el buffer ordenadas por las de mayor longitud.

Principales Funciones

	head	Nombre	Tipo	Descripción
Argumentos	Argumento 1	**N**	**int**	**Cantidad de líneas que se quieren mostrar.**
Variables Locales	Variable 1	**i**	**int**	**Contador de líneas.**
	Variable 2	**Buffer[LEN]**	**char**	**Líneas de caracteres que se guardan y muestran.**
Valor Devuelto			**char**	**muestra por pantalla las primeras líneas.**
Descripción de la Función				**Esta función lee todas las líneas de la entrada estándar con buffer hace un bucle con tope n y muestra las n primeras líneas introducidas.**

	insertarFinal	Nombre	Tipo	Descripción
Argumentos	Argumento 1	**Caracter[LEN]**	**char**	**Línea que se ha introducido**
Variables Locales	Variable 1	***nodo**	**Lista**	**Puntero nuevo a lista**
	Variable 2	***aux**	**Lista**	**Puntero auxiliar para realizar movimientos de orden**
Valor Devuelto			**caracter**	**Lista modificada**
Descripción de la Función				**implementación de listas. insertar final primero aux lee la lista e inserta al final de esta la última línea introducida.**

	insertaOrdenado	Nombre	Tipo	Descripción
Argumentos	Argumento 1	**Carácter[LEN]**	**char**	**línea introducida de longitud LEN = 1024**
Variables Locales	Variable 1	***nodo**	**Lista**	**Puntero a lista**
	Variable 2	***aux**	**Lista**	**Puntero auxiliar para realizar movimientos de orden**
	Variable 3	***aux2**	**Lista**	**Puntero auxiliar para realizar movimientos de orden y no perder elementos.**
Valor Devuelto			**Lista**	**Lista ordenada.**
Descripción de la Función				**ordena las líneas de mayor a menor usan haciendo uso de los dos puntos de los auxiliares e incluye la nueva línea de forma ordenada**

	mostrarP	Nombre	Tipo	Descripción
Argumentos	Argumento 1		**void**	
Variables Locales	Variable 1	***aux**	**Lista**	**Puntero auxiliar**
	Variable 2	***aux2**	**Lista**	**Puntero auxiliar**
Valor Devuelto			**Lista**	**Lista por pantalla**
Descripción de la Función				**mientras que el puntero no apunte a null imprime cadena apunta siguiente y libera al puntero auxiliar.**

	Tail	Nombre	Tipo	Descripción
Argumentos	Argumento 1	**N**	**int**	**Cantidad de líneas que se quieren mostrar.**
Variables Locales	Variable 1	**contL**	**int**	**Contador de líneas.**
	Variable 2	**contP**	**int**	**Contador de las veces que se muestra.**
	Variable 3	**Buffer[LEN]**	**char**	**Línea guardada o leída.**
	Variable 4	***aux**	**Lista**	**Puntero auxiliar que recorre la lista para añadir líneas, o guardar líneas.**
	Variable 5	***aux2**	**Lista**	**Puntero auxiliar que recorre la lista para añadir líneas, o guardar líneas.**
Valor Devuelto			**Lista**	**muestra por pantalla las n últimas líneas introducidas.**
Descripción de la Función				**haciendo uso de un bucle para contar líneas, compararlas a el máximo n, para sabes si mostrar todas, y otro para mostrar las últimas.**

	longlines	Nombre	Tipo	Descripción
Argumentos	Argumento 1	**n**	**int**	**Cantidad de líneas que se quieren mostrar.**
Variables Locales	Variable 1	**contL**	**int**	**Contador de líneas.**
	Variable 2	**contP**	**int**	**Contador de las veces que se muestra.**
	Variable 3	**Buffer[LEN]**	**char**	**Línea guardada o leída.**
	Variable 4	***aux**	**Lista**	**Puntero auxiliar que recorre la lista para añadir líneas, o guardar líneas.**
	Variable 5	***aux2**	**Lista**	**Puntero auxiliar que recorre la lista para añadir líneas, o guardar líneas.**
Valor Devuelto			**Lista**	**muestra por pantalla las n líneas más largas.**
Descripción de la Función				**cuenta líneas haciendo uso de un bucle y las ordena con la función de insertaOrdenada de mayor o menor.**

Casos de Prueba

Se hicieron distintas pruebas utilizando test.c las cuales se probaba los argumentos si admitía más de 3 o menos de 1 en los cuales saltaba el mensaje de error.

Posteriormente se probó La introducción de menos líneas de lo permitido, más de lo permitido, caracteres y teclas especiales. Los cuales se comprobó la salida de un mensaje de error y se ponía por defecto 10 como máximo.

También se aprobó la introducción de caracteres y palabras en la selección de la función. Los cuales se probó la salida de un mensaje de error que te muestra las opciones permitidas.

También se realizaron pruebas dedicadas a cada función:

En head se probó que mostrará las líneas introducidas.

En tail se probó que mostrarse las últimas líneas introducidas.

Y en Longlines se probó que se mostrasen las líneas más largas.

Comentarios Personales

Me gustaría aclarar que esta practia se ha realizado en solitario, también recalcar la ayuda de compañeros, profesores y bibliotecas virtuales para la realización de esta práctica. Con ayuda de páginas web como Stack Overflow y muchas otras.

Como primer comentario me gustaría enfatizar en la dificultad de adaptación a un lenguaje distinto de los últimos estudiados como podrían ser Java o Python, c es bastante más restrictivo en su sintaxis que estos dos anteriores lo cual me ha complicado bastante el uso de este lenguaje y sobre todo las máquinas virtuales de Ubuntu y los fallos que conllevan.

Por otra parte, con esta práctica he sido capaz de aprender el lenguaje c y hacer uso de funciones básicas, debido a su parecido con Pascal

Al haber decidido implementar una lista en vez de un array me ha complicado un poco el uso de punteros y bucles, pero lo he realizado así para aprender a usar listas en este lenguaje.

Un cambio que sugerir, seria hacer uso de otro compilador más sencillo e intuitivo sin tener que depender de myapps como obligación, debido a sus fallos he tenido que hacer uso de otros compiladores y máquinas virtuales y podría verse afectada la nota por este motivo, también me gustaría la dedicación de una fracción de una clase a la instalación de una máquina virtual.

A esta práctica se le habrá dedicado un tiempo que rondará las 30 horas debido a realizarla en solitario y tener que hacer uso de varias fuentes distintas para lograr su funcionamiento.