Sistemas Operativos - Práctica 1

José Javier Meiriño Migens

Índice

Índice **2**

int head(int n) **3**

Descripción: 3

Comentario: 3

int tail(int n) **3**

Descripción: 3

Comentario: 4

int longlines(int n) **4**

Descripción: 4

Comentario: 4

Notas adicionales **5**

int head(int n)

Descripción:

El código inicializa el array estático Buffer a 1024 bytes, y se entra en un bucle While donde, en cada iteración, se lee de la entrada estándar, se le asigna a buffer lo leído, y se imprime. Si se supera las n lecturas, o se lee un caracter nulo, el bucle finaliza acabando también la función.

Comentario:

La función Head fue muy fácil de implementar, ya que se hizo un ejercicio en clase sobre implementar la función en forma de mandato. Por lo tanto, la implementación de esta función simplemente es la versión que se hize en clase pero usando el parámetro N en vez de comprobar los argumentos del mandato y usar un número predefinido en caso de que no se pasase dicho argumento.

En esta parte no he hecho uso de la memoria dinámica, ya que simplemente se muestra por pantalla la línea leída.

int tail(int n)

Descripción:

Se reserva la memoria con malloc, como n veces el tamaño de un puntero a char y si devolviese NULL (Memoria llena, o cualquier otro error), entonces la función finaliza devolviendo 1.

El bucle While comprueba si se ha leído algo no nulo, y si se han leído más de n líneas, se desplazan los elementos del array a la derecha y se inserta por el final. En caso contrario, sólo se inserta por el final.

Comentario:

La función tail fue más complicada de implementar. Esta función ya exige el uso de memoria dinámica para almacenar las últimas líneas leídas desde la entrada estándar, y por lo tanto hay que hacer uso de malloc/free. El uso de la memoria dinámica resultó bastante sencillo una vez claro el concepto de los dobles punteros.

La mayor dificultad fue la forma de añadir las líneas leídas. Intenté el uso de la operación módulo (%) para tener la posición en donde colocar la siguiente línea, pero solo funcionaba

para cuando el usuario introdujese < n líneas. Posteriormente procedí a implementar a un algoritmo donde

- Si se introducen > n líneas, se realiza un bucle for desplazando las líneas a la izquierda y se inserta en la última posición
- En cualquier otro caso, se inserta en la siguiente posición libre

Finalmente, cuando se termina de leer, se escriben las líneas resultantes al mismo tiempo que se libera su memoria.

int longlines(int n)

Descripción:

Similar a tail(int n), en esta función se reserva memoria con malloc y se comprueba si no se ha producido un error en ello.

Posteriormente, se entra en un bucle While que comprueba si la longitud de la cadena leída es superior al elemento i. En caso de ser así, se desplazan todos los elementos desde ese punto, a la derecha, se inserta la línea leída en la posición i y se sale del bucle.

Cuando se lea una línea nula, la función finaliza mostrando todos los elementos del array y liberando la memoria reservada.

Comentario:

La función longlines fue similar a la función tail en que lo más complicado de su implementación fue la forma de introducir las líneas (o más bien, sus punteros) en el array de punteros.

El algoritmo pensado para esto, fue uno donde se inicializan los valores a NULL y se va recorriendo uno a uno los elementos de la lista desde el principio, y

- Si esa posición está vacía (NULL) o la longitud de la línea es igual o menor a la que se ha leído, entonces desplazan todas las líneas desde ese punto a la derecha, se inserta en la posición i, y se sale del while con i = n
- En otro caso, el bucle continúa hasta el final comparando la longitud de las líneas existentes

Finalmente, se muestran por pantalla las líneas guardadas y se va liberando la memoria ocupada.

Notas adicionales

Al compilar con GCC y el argumento “-ansi”, la función strdup daba warnings sobre que la función estaba sin definir y debía incluir <string.h>, pese a que ya estaba hecho. Por lo tanto, decidí redefinir la función y así evitar dichos warnings.

Finalmente, en la función longlines, la instrucción free(lineas), similar a la de la función tail, producía un error en tiempo de ejecución, por lo que decidí eliminarla.

Conclusiones

Como conclusión a esta práctica, se obtiene que ha sido un buen ejemplo de prueba para poner en práctica los conceptos de reserva/liberación de memoria, entrada estándar y redireccionamiento.