

CC3301 Programación de Software de Sistemas – Semestre Primavera 2022 – Tarea 4 – Prof.: Luis Mateu

El archivo de texto *cola.txt* almacena una cola de prioridad en el formato que muestra la columna de la izquierda de la siguiente tabla:

archivo <i>cola.txt</i>	Ejemplo de uso	<i>cola.txt</i> después
4 pedro 8 juan 2 diego 7 ximena 4 ana 1	\$./extraer.bin cola.txt juan \$	3 pedro 8 ximena 4 diego 7 ximena 4 ana 1

La primera línea del archivo tiene 4 caracteres más el $\backslash n$ (cambio de línea) y corresponde a la cantidad de elementos almacenados en la cola. Cada una del resto de las líneas tiene 16 caracteres más el $\backslash n$. Los primeros 10 corresponden a un texto almacenado en la cola y los 6 restantes a su prioridad. En el ejemplo la mejor prioridad es 2. El archivo podría contener 0, 1 o más líneas sobrantes al final (de 16+1 caracteres) debido a elementos extraídos, pero solo al final. Su contenido es irrelevante.

Programe el comando *extraer* que recibe como único argumento el nombre del archivo que contiene la cola y extrae de ella el texto con la mejor prioridad, mostrándolo en pantalla. Para la extracción Ud. **debe** (i) leer secuencialmente todos los elementos de la cola, buscando el texto con mejor prioridad; (ii) mostrarlo en pantalla; (iii) **usar *fseek* para actualizar el tamaño de la cola**; y (iv) **usar *fseek* para mover el último elemento de la cola a la línea que queda libre**. Para (ii) recuerde que al mostrar el texto con *printf*, ese texto debe ser un string y por lo tanto debe terminar con $\backslash 0$. En el archivo el texto no está terminado con $\backslash 0$. Además al mostrar el texto debe incluir los espacios en blanco y por lo tanto siempre deberá mostrar 10 caracteres.

La columna del centro de la tabla de arriba muestra un ejemplo de uso del comando *extraer* y la columna de la derecha muestra el contenido de *cola.txt* después de la extracción. La salida estándar de *./extraer.bin* es el texto *juan* seguido de 6 espacios en blanco, porque el texto contiene 10 caracteres siempre. Observe que el i -ésimo elemento de la cola está en la posición $4+1+i*(16+1)$ en el archivo (partiendo con $i \equiv 0$). Está prohibido almacenar todo el contenido de la cola en la memoria porque podría ser demasiado grande.

Si la cola está vacía Ud. debe diagnosticar el error en la salida estándar de errores con este mensaje:

error: cola.txt esta vacia

Si el archivo no existe, Ud. debe diagnosticar el error en la salida estándar de errores usando la función *perro*.

La salida estándar debe estar vacía en caso de detectar cualquier tipo de error.

Ayuda: Estas funciones le serán de utilidad.

```
int n= atoi("25 \n"); // n es 25
char s[6], d[6];
sprintf(s, "%-4d\n", n); // imprime en un string, no en un archivo,
// %-4d significa mostrar entero en 4 caracteres, alineado a la izquierda,
// s es el string "25 \n" de largo 5
memcpy(d, s, 4); // copia 4 bytes a partir de s en d
```

Instrucciones

Descargue *t4.zip* de U-cursos y descomprímalo. Ejecute el comando *make* sin parámetros en el directorio *T4* para recibir instrucciones acerca del archivo en donde debe programar su solución (*extraer.c*), cómo compilar, probar y **depurar** su solución, y los requisitos que debe cumplir para aprobar la tarea.

Entrega

Ud. solo debe entregar por medio de U-cursos el archivo *extraer.zip* generado por el comando *make zip*. **A continuación es muy importante que descargue de U-cursos el mismo archivo que subió, luego descargue nuevamente los archivos adjuntos y vuelva a probar la tarea tal cual como la entregó.** Esto es para evitar que Ud. reciba un 1.0 en su tarea porque entregó los archivos equivocados. Créame, sucede a menudo por ahorrarse esta verificación. Se descontará medio punto por día de atraso. No se consideran los días de receso, sábado, domingo o festivos.