

TAREA 1 - ELO320 ESTRUCTURA DE DATOS Y ALGORITMOS
PROF.: NICOLÁS GÁLVEZ R.
AYUDANTES: PABLO SANCHEZ - RODRIGO LOBOS - GABRIEL ZAPATA
CAMPUS SAN JOAQUÍN, UTFSM. 08 DE ABRIL DE 2023

1 USM-QP:

A nivel internacional, ha habido una filtración de la app de mensajería *WazaaaApp*. En esta se pueden identificar variadas conversaciones o *chat* de celebridades, tales como Shatira o Tonga Clockicic. Los productores del programa de farándula universitaria USM-QP, le han solicitado a los estudiantes de ELO-320 de la UTFSM, crear un programa para poder fisgar en cada *chat* y analizar su contenido. Las filtraciones contienen conversaciones de los famosos con sus amigos, familiares y colegas, incluyendo discusiones, planes de trabajo y opiniones sobre temas polémicos. Usted deberá crear un programa que permita mostrar las conversaciones por pantalla, utilizando ventanas de tiempo según las solicitudes del usuario. Además, se indican las siguientes consideraciones:

- El programa contempla el uso de dos (2) archivos de texto: uno que contiene el contenido de alguno de los *chat* filtrados y otro relacionado a una lista de usuarios de la aplicación *WazaaaApp* cuyas conversaciones fueron filtradas.
- El programa cargará la conversación completa entre dos usuarios mediante la terminal, iniciando con los últimos 10 mensajes ordenados de forma cronológica. Luego, se podrá navegar a través de la conversación mediante una **ventana deslizante**, pero siempre mostrando 10 mensajes ordenados de forma cronológica.
- La ventana de mensajes puede desplazarse avanzando o retrocediendo cronológicamente siempre mostrando 10 mensajes ordenados por dicho criterio.

Cada *chat* filtrado estará almacenado en un archivo de texto plano cuyo nombre tendrá el formato `user1-user2-year.wzp`, donde `user1` y `user2` corresponden a los usuarios de *WazaaaApp* que entablaron la conversación, y `year` es el año en que inició ésta. Un ejemplo es el archivo `Shatira-Pike-2023.wzp`, el cual posee un formato definido, tal como se muestra a continuación:

`Shatira-Pike-2023.wzp`

```
...
[07/01/2023 3:24 PM] Shatira +34 612 34 56 78: Sorry, baby, hace rato, que yo debí botar ese gato
[07/01/2023 3:26 PM] Piké +34 612 34 56 79: ¿Qué gato? ¿De qué estás hablando, Shatira? ¿Por qué me ←
estás mandando esto?
[07/01/2023 3:28 PM] Shatira +34 612 34 56 78: Una loba como yo no está pa' novato '
[07/01/2023 3:28 PM] Shatira +34 612 34 56 78: Una loba como yo no está pa' tipos como tú—uh—uh—uh—←
uh
[07/01/2023 3:28 PM] Shatira +34 612 34 56 78: Pa' tipos como tú—uh—uh—uh—uh
[07/03/2023 9:00 AM] Piké +34 612 34 56 79: Hola Shatira, no sé si leíste los comentarios de la ←
canción que publicaste en Youtube, pero son muy crueles. ¿Cómo pudiste hacer eso?
[07/03/2023 9:05 AM] Shatira +34 612 34 56 78: Piké, ¿cómo pudiste engañarme durante tanto tiempo? ←
Esa canción fue mi forma de expresar mi dolor y mi enojo. No te justifiques.
[07/03/2023 9:07 AM] Piké +34 612 34 56 79: No estoy tratando de justificarme, Shatira. Solo no ←
entiendo cómo pudiste ser tan cruel.
[07/03/2023 9:10 AM] Shatira +34 612 34 56 78: No es crueldad, Piké. Es mi forma de lidiar con lo ←
que pasó. Necesitaba sacarlo de alguna manera y la canción fue mi escape.
...
```

Considere que los puntos suspensivos indican que el archivo contiene mucho más contenido. Además, cada línea de este archivo tiene el formato `[fecha hora] usuario fono: mensaje`, en donde:

- Fecha: fecha cuando se emitió el mensaje, en formato `dd/mm/aaaa` (día/mes/año).
- Hora: hora del día en que se emitió el mensaje, en formato `hh:mm AM/PM` (horas:minutos AM ó PM).
- Usuario: es el usuario de *WazaaaApp* que emitió el mensaje. Largo máximo: 25 caracteres.
- Fono: es el número de teléfono móvil del usuario que emitió el mensaje.

- Mensaje: el mensaje que el usuario emitió. String dinámico, debe manejarlo mensaje a mensaje para un uso correcto de la memoria. Recuerde que algunos caracteres del español tienen dos bytes (tildes, inicio de interrogación o exclamación, etc.).

El archivo que contiene la información de los usuarios de *WazaaaApp* que han sido afectados por esta filtración, se llama *user.wzp* y se muestra a continuación:

user.wzp

```
...
+56 9 6543 2109
+1 555 123 4567
+33 6 12 34 56 78
+34 612 34 56 78
+56 9 4321 5678
...
+52 55 1234 5678
+34 612 34 56 79
+56 9 5432 1098
...
```

Este archivo contiene un número de teléfono por línea, vale decir, el número de teléfono móvil de cada afectado. Considere que los puntos suspensivos indican que existen más teléfonos móviles de los que se muestran.

Para mostrar las conversaciones, cuyo tamaño puede ser muy largo, se utilizará una **ventana deslizante** para navegar a través del *chat* filtrado. La idea conceptual de ésta se muestra en la Figura 1 para una ventana deslizante de tres (3) mensajes. En ésta se aprecia en color negro los mensajes que están en la ventana deslizante y, por lo tanto, se muestran por pantalla. Además, de que esta ventana puede ir deslizándose hacia arriba o hacia abajo (*scroll up* y *scroll down*) de forma cronológica en la conversación.

tonga-horived-2022.wzp	tonga-horived-2022.wzp
<pre>[20/07/2022 3:10 PM] Horived +56 9 1234 5678: Sí↵ , mi amor. Y todo gracias a nuestro esfuerzo↵ y dedicación. Juntos somos imparables. [20/07/2022 3:15 PM] Tonga +56 9 8765 4321: ↵ Totalmente de acuerdo, mi amor. Y sé que ↵ vamos a lograr cosas aún más grandes en el ↵ futuro. [20/07/2022 3:20 PM] Horived +56 9 1234 5678: As↵ í es, mi amor. No hay límites para lo que ↵ podemos lograr juntos. [18/09/2022 10:15 AM] Horived +56 9 1234 5678: ↵ Amor, te tengo que contar algo muy grave. [18/09/2022 10:20 AM] Tonga +56 9 8765 4321: ¿Qu↵ é sucedió, Horived? ¿Estás bien? [18/09/2022 10:25 AM] Horived +56 9 1234 5678: ↵ Estela, nuestra socia en la compraventa de ↵ relojes de oro, ha sido detenida por la ↵ policía. La acusan de estafa y venta de artí↵ culos robados. [18/09/2022 10:30 AM] Tonga +56 9 8765 4321: ¿Có↵ mo puede ser posible? ¿No chequeamos que ↵ todo fuera legal? [18/09/2022 10:35 AM] Horived +56 9 1234 5678: ↵ Lo hice, pero creo que se las arregló para ↵ engañarnos. Parece que ha estado operando ↵ con relojes robados todo el tiempo. [18/09/2022 10:40 AM] Tonga +56 9 8765 4321: No ↵ puedo creerlo. ¿Qué vamos a hacer ahora? [18/09/2022 10:45 AM] Horived +56 9 1234 5678: ↵ Primero debemos asegurarnos de que no ↵ estemos implicados en nada ilegal. Después ↵ de eso, tendremos que buscar una nueva ↵ manera de invertir nuestro dinero.</pre>	<pre>[20/07/2022 3:10 PM] Horived +56 9 1234 5678: Sí↵ , mi amor. Y todo gracias a nuestro esfuerzo↵ y dedicación. Juntos somos imparables. [20/07/2022 3:15 PM] Tonga +56 9 8765 4321: ↵ Totalmente de acuerdo, mi amor. Y sé que ↵ vamos a lograr cosas aún más grandes en el ↵ futuro. [20/07/2022 3:20 PM] Horived +56 9 1234 5678: As↵ í es, mi amor. No hay límites para lo que ↵ podemos lograr juntos. [18/09/2022 10:15 AM] Horived +56 9 1234 5678: ↵ Amor, te tengo que contar algo muy grave. [18/09/2022 10:20 AM] Tonga +56 9 8765 4321: ¿Qu↵ é sucedió, Horived? ¿Estás bien? [18/09/2022 10:25 AM] Horived +56 9 1234 5678: ↵ Estela, nuestra socia en la compraventa de ↵ relojes de oro, ha sido detenida por la ↵ policía. La acusan de estafa y venta de artí↵ culos robados. [18/09/2022 10:30 AM] Tonga +56 9 8765 4321: ¿Có↵ mo puede ser posible? ¿No chequeamos que ↵ todo fuera legal? [18/09/2022 10:35 AM] Horived +56 9 1234 5678: ↵ Lo hice, pero creo que se las arregló para ↵ engañarnos. Parece que ha estado operando ↵ con relojes robados todo el tiempo. [18/09/2022 10:40 AM] Tonga +56 9 8765 4321: No ↵ puedo creerlo. ¿Qué vamos a hacer ahora? [18/09/2022 10:45 AM] Horived +56 9 1234 5678: ↵ Primero debemos asegurarnos de que no ↵ estemos implicados en nada ilegal. Después ↵ de eso, tendremos que buscar una nueva ↵ manera de invertir nuestro dinero.</pre>

Figure 1: Ventana Deslizante de tres mensajes.

Para realizar el programa solicitado, USM-QP le pidió al profesor de ELO320 diseñar un archivo base para desarrollar el programa, el header (`usmqp.h`). Además, el profesor les indica que puede agregar cosas si ud. estima conveniente, pero **debe usar y no puede modificar** todo lo que se encuentra ya definido.

`usmqp.h`

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

typedef struct user{
    char * telefono; // string: teléfono móvil del usuario
    char * nombre; // string: nombre del usuario
    struct user * sgte; // puntero: siguiente.
} contacto;

typedef struct WazaaaApp{
    char * fecha; // string: fecha de mensaje
    char * hora; // string: hora de mensaje
    contacto * emisor; // puntero a emisor del mensaje
    char * contenido; // string: contenido del mensaje
    struct WazaaaApp * sgte; // puntero: siguiente.
} mensaje;
```

En este header se encuentra la estructura base para poder tratar la información de los archivos necesarios para que el programa funcione. Una vez entregada la información relevante, el profesor les asigna las siguientes labores:

- (10 puntos) Leer el archivo de texto base de los contactos con conversaciones filtradas en *WazaaaApp*, obtener la información necesaria para ser almacenada en una Estructura de Datos de **su elección**. Considere que el nombre del archivo será obligatoriamente un parámetro del ejecutable.
- (10 puntos) Leer el archivo de texto base de la conversación filtrada, y obtener la información necesaria para ser almacenadas en una **pila**. Considere que el nombre del archivo será obligatoriamente un parámetro del ejecutable.
- (15 puntos) Crear un archivo, llamado `contactos.c`, en donde implemente la interfaz de la Estructura de Datos escogida para almacenar la información de los contactos de *WazaaaApp*. Esta estructura de datos debe usar obligatoriamente la `struct user` como base. Considere que la interfaz debe ser capaz, al menos, de:
 - Añadir un nuevo contacto a la EDA.
 - Eliminar un contacto de la EDA.
 - Buscar la ubicación de un contacto en la EDA.
 - Vaciar la EDA.
- (15 puntos) Crear un archivo, llamado `pila.c`, en donde implemente la interfaz de la pila que almacenará los mensajes del *chat* filtrado. Considere la interfaz debe ser capaz, al menos, de:
 - Push o añadir elemento en la pila.
 - Pop o eliminar un nodo de la pila.
 - Peek o mostrar primer elemento de la pila.
 - vaciar Pila.
- (30 puntos) Crear un archivo, llamado `Ventana.c`, en donde implemente la interfaz de la **ventana deslizable**. Para lograr este objetivo puede usar TDAs complementarios a las EDAs obligatorias. Comente en su código claramente el proceso que está diseñando.
 - Imprimir los mensajes en la ventana deslizable.
 - Verificar si está el inicio o final de la conversación.
 - Avanzar y retroceder, por separado, entre los mensajes de la conversación.

- Buscar un mensaje que contenga una palabra en específico.
- (f) (20 puntos) Crear un programa que permita cargar la lista de usuarios con conversaciones filtradas en *Waza-aaApp* y una conversación filtrada. El programa debe mostrar por pantalla los mensajes de la ventana deslizante y dar al menos las siguientes opciones:
- Avanzar un mensaje en la conversación (**scroll down**).
 - Retroceder un mensaje en la conversación (**scroll up**).
 - Ir con la ventana deslizante al sector de la conversación que contenga dentro de su contenido un mensaje con una palabra específica ingresada por el usuario.
 - Cerrar el programa.

2 REGLAS DE ENTREGA Y CONSIDERACIONES GENERALES

- Este trabajo debe realizarse **individualmente**, vale decir, en **grupos de un (1) estudiante**. No se harán excepciones.
- El programa debe ser desarrollado en lenguaje C, y compilado con la versión de gcc disponible en el servidor Aragorn¹: gcc 4.8.5.
- No hay un límite máximo de funciones a realizar, si ud. desea modularizar al máximo, tiene la libertad de hacerlo.
- La tarea debe ser entregada en la plataforma **AULA USM**, el día Jueves 18 de Mayo de 2023 hasta las 23:59:59 Hora de Chile Continental (UTC -4).
- Cada minuto de atraso, implicará un descuento siguiendo la sucesión de Fibonacci. Así, diez minutos de atraso implicarían un descuento de cincuenta y cinco (55) puntos ($f(10) = 55$).
- La tarea debe incluirse en un archivo comprimido **.tar.gz**. El nombre del archivo debe seguir la siguiente estructura: **tarea1-eda-nombre-apellido-paralelo.tar.gz**; e.g., **tarea1-eda-oliver-atom-p200.tar.gz**.
- El archivo comprimido debe incluir, al menos, lo siguiente:
 - Cabecera **.h**: Archivo cabecera en el cual se deben incluir todas las bibliotecas a usar en el programa, además de las definiciones de macros, variables globales, tipos de datos personalizados, struct y prototipos de todas las funciones.
 - Código **.c**: Código del programa solicitado. Considere que éste es un desafío de al menos tamaño mediano, por lo tanto es recomendable dividir el código en diferentes archivos agrupados por su finalidad.
 - **README**: Archivo de texto plano en el cual se debe incluir: (1) una pequeña reseña del programa, (2) las condiciones de compilación y ejecución, (3) las instrucciones de compilación y ejecución, y (4) la información del creador del sistema.
 - **Makefile**: Archivo de compilación automática del sistema. Una cápsula de video para su confección se encontrará disponible en AULA USM.
- La revisión de los programas se hará utilizando diferentes archivos de tamaño variable. Por lo tanto, debe programar de forma genérica para cualquier cantidad de información de entrada.
- Cada fuga de memoria será penalizada. Utilice **valgrind** para verificar la correcta asignación, uso y liberación de memoria en la ejecución de su programa.
- Si su programa no compila, su nota será automáticamente **un cero (0)**.
- Cualquier atisbo de copia, será penalizada con máxima severidad.
- No deje la tarea para último momento, planifique bien sus tiempos y constrúyala paso a paso.
- Consulte sus dudas, lo más pronto posible, a través de los diversos medios de la asignatura.

¹ssh aragorn.elo.utfsm.cl -l cuentausm