

TICS311: Tarea #1

Universidad Adolfo Ibañez

Danilo Bórquez Paredes
danilo.borquez.p@uai.cl

Miguel Rodríguez
miguel.rodriguez.s@edu.uai.cl

Miguel Romero
miguel.romero.o@uai.cl

21 de febrero de 2022

Objetivos

- Primer acercamiento a Programación en C
- Manejo de archivos y funciones básicas

1. Enigma

Cuenta la historia que durante la segunda guerra mundial existía un bando que se comunicaba con mensajes cifrados a través de ondas de radio. De esta manera, el bando enemigo podía interceptar el mensaje, pero no entendería lo que en él estaba escrito. A menos claro, que tuviera la llave que decodificara el mensaje...

Es así como un gran matemático llamado Alan Turing¹ logró desarrollar una máquina (Enigma) que encontraba ciertos patrones en los mensajes enemigos, llegando así a decodificarlos. Turing es considerado uno de los padres de la ciencia de la computación.

Varios años más tarde, usted es contratado para crear una máquina que sea capaz de codificar o decodificar un mensaje que se encuentra dentro de un archivo de texto, utilizando dos palabras claves. La mecánica que describe la codificación es:

- Dos palabras son utilizadas como clave.
- Ambas palabras clave deben tener igual largo.
- Las letras contenidas en las palabras claves deben ser **TODAS DIFERENTES**.
- Las palabras clave pueden contener cualquier caracter (incluido el espacio). Por ejemplo se podría usar **,!a**, que tiene tres caracteres.
- Cada letra del documento que se encuentra en la primera palabra clave es sustituida por la letra que se encuentra en la misma posición de la segunda palabra clave.
- No puede haber ninguna letra repetida entre las dos palabras claves.

La Tabla 1 muestra un ejemplo de dos palabras claves: **puro** y **hace**. Cada letra del mensaje de entrada que sea igual a una P, es cambiada por una H. El caso contrario también es correcto: Cada letra del mensaje de entrada que sea igual a H **TAMBIÉN ES CAMBIADA A UNA P**.

Cuando se decodifica un mensaje el procedimiento es exactamente el mismo que para decodificar.

¹https://en.wikipedia.org/wiki/Alan_Turing

	0	1	2	3
palabra clave 1	P	U	R	O
palabra clave 2	H	A	C	E

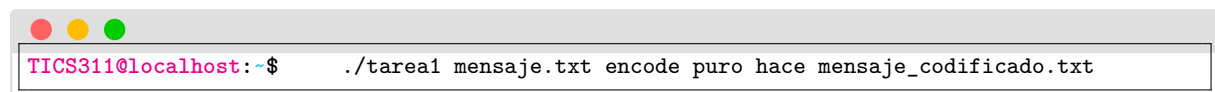
Cuadro 1: Ejemplo de dos palabras clave

1.1. Entrada del programa

Su programa recibirá como entrada 5 parámetros, en el siguiente orden:

1. nombre del archivo con el mensaje de entrada.
2. La palabra “encode” si lo que quiere es codificar el mensaje, o la palabra “decode” si lo que quiere es decodificar el mensaje.
3. La palabra clave 1.
4. La palabra clave 2.
5. El nombre del archivo de salida (Ingresado por el usuario).

A continuación un ejemplo de ejecución:



```
TICS311@localhost:~$ ./tarea1 mensaje.txt encode puro hace mensaje_codificado.txt
```

El nombre del archivo (mensaje.txt) es sólo un ejemplo, dado que puede llamarse de cualquier manera. A continuación un ejemplo de este archivo, y su versión codificada:

mensaje.txt
H0la a todos!

mensaje_codificado.txt
PElu u tedes!

Mucha atención a las mayúsculas y minúsculas. Las palabras clave pueden ser ingresadas en mayúsculas o minúsculas. A su vez, si en el mensaje hay mayúsculas y minúsculas, estas deben mantener ese estado. En el ejemplo, todas las letras “o” fueron cambiadas por E mayúsculas o minúsculas, según sea el caso.

1.2. Salida del programa

El programa debe generar un archivo con el nombre ingresado por el usuario. A su vez debe mostrar por pantalla si el archivo fue codificado o decodificado.

En caso que el ingreso de parámetros no sea el correcto, entonces el programa no debe ejecutarse y debe mostrarle al usuario una frase indicando el error. Las validaciones que debe hacer el programa son a lo menos las siguientes:

- El programa debe recibir 5 argumentos.
- El segundo argumento debe ser la palabra “encode” o “decode”.
- El largo de las dos palabras clave deben ser iguales.
- Los nombres del archivo de entrada y salida deben ser diferentes.

A su vez, el nombre del archivo de entrada siempre existirá (no es necesario que lo valide).

2. Sobre la entrega

- La tarea debe ser hecha en lenguaje de Programación C.
- Cada grupo puede ser de 2 o 3 personas.
- La tarea se debe entregar el día **Domingo 20 de Marzo** a las 23:59.
- Por cada día de atraso se descontará 1 punto, comenzando a las 00:00 horas del siguiente día. Por ejemplo si entrega la tarea a las 00:00 del siguiente día, la nota máxima que puede obtener es un 6.0
- Para la corrección se utilizará un compilador gcc v 5.1 o superior
- La entrega se realiza por la plataforma Webcursos²
- El archivo a entregar debe ser un zip que contenga una carpeta en su interior (y sólo una carpeta) con el nombre **tarea1**. Dentro de esa carpeta debe haber un Makefile³ y por lo menos un main.c. Además el nombre de su zip debe ser grupoX-tarea1.zip, donde X es el número de su grupo.
- **IMPORTANTE:** el directorio **tarea1** debe contener el Makefile que generará el archivo ejecutable **tarea1**.

²<http://webcursos.uai.cl>

³Este archivo deben generarlo. Información útil pueden encontrarla en <https://stackoverflow.com/questions/1484817/how-do-i-make-a-simple-makefile-for-gcc-on-linux>. Pueden probar su Makefile en un computador con MAC o Linux, o en VSCode en la terminal con UBUNTU. También puede probarse en la shell de <https://repl.it/>