



**Condiciones de entrega de la parte práctica:**

1. Horario y duración:
  - a. La duración es de 180 minutos.
  - b. La hora de inicio es 19.00 y la de finalización es 23.00
  - c. Todas las entregas realizadas fuera de término no serán tenidas en cuenta para su corrección
2. Entrega:
  - a. Se utilizará el repositorio individual utilizado para la entrega de los trabajos prácticos
  - b. Todos los archivos .c y .h deben ser subidos a una carpeta con el nombre 2P
  - c. Cada ejercicio debe tener un main que demuestre el funcionamiento de la función solicitada el nombre de cada archivo debe respetar el siguiente formato 2P\_ejercicioNumero.c Por ejemplo para el ejercicio 1 el nombre del archivo que contiene el main es 1P\_01.c
  - d. Todas las funciones deberán ser colocadas en un archivo con el nombre 2P\_funciones.c
  - e. Se recomienda hacer un commit cada 30 minutos.
  - f. Al finalizar el parcial debe hacer el commit al repositorio y subirlo a la tarea asignada en el campus virtual.
  - g. Complete los encabezados de los archivos y las funciones. Indicando en la descripción del archivo la forma de compilar y linkear el código.

**Parte práctica:**

1. (2.5 Puntos) Implemente una función que divida un archivo en varios archivos del tamaño indicado. El prototipo de la función es el siguiente

```
int dividirArchivos(char *nombre, int tamano)
```

Donde:

- nombre: Nombre del archivo a dividir.
- tamano: Tamano de cada archivo resultante de la división.

Devuelve: La cantidad de archivos en los que se dividió el original o un número negativo indicando el error.

- -1: Si no pudo abrir el archivo origen
- -2: Si la cantidad de archivos resultantes es mayor que 999
- -3: Si la cantidad de archivos resultantes es cero

El nombre de los archivos resultantes serán numerados luego de la extensión, por ejemplo si el nombre del archivos original es prueba.txt y este se dividirá en 4 partes, los nombres de los archivos resultantes serán los siguientes prueba.txt.000; prueba.txt.001; prueba.txt.002; prueba.txt.003;

2. (2.5 Puntos) Realice un programa que implemente un servidor TCP que reciba strings de 128 caracteres (incluye el '\0') los codifique en ROT-13 y se lo devuelva al cliente. Cada uno de los string recibidos deberán ser almacenados en un archivo a razón de un string por línea.

El port del servidor junto con el nombre de archivo de log deberá pasarse como argumento de main  
Ejemplo

- Servidor que escucha en el port 5000 y almacena los string en un archivo con el nombre log.txt
- ```
./2P_02.out 5000 log.txt
```

El algoritmo ROT-13 consiste en sustituir cada letra por una que se encuentra trece posiciones por delante. Por ejemplo la A se reemplaza por la N, la B por la O y así sucesivamente. Para las últimas trece letras la secuencia se invierte. A continuación se muestra la tabla de las equivalencias entre las letras

|   |   |   |   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | B | C | D | E | F | G | H | I | J | K | L | M |
| N | O | P | Q | R | S | T | U | V | W | X | Y | Z |

Ejemplo:

|                  |   |   |   |   |  |   |   |   |   |   |
|------------------|---|---|---|---|--|---|---|---|---|---|
| Texto sin cifrar | H | O | L | A |  | M | U | N | D | O |
| Texto cifrado    | U | B | Y | N |  | Z | H | A | Q | B |

3. (2 Puntos) Implemente un programa que lea strings desde teclado y los transmita a un servidor TCP luego se quede esperando a que le responda un string el cual estará encriptado en ROT-13 e imprima en pantalla. La IP y port a la cual se conectará el cliente TCP se pasará por argumentos de main.

Ejemplo

- Cliente se conecta a la IP 127.0.0.1 y el port 5000  
`./2P_03.out 127.0.0.1 5000`

4. (1 Puntos) Implemente un makefile que realice las siguientes operaciones:
- a. Borre todos los archivos generados (clean)
  - b. Compile, linkee y genere los ejecutables de los ejercicios 2 y 3
  - c. Haga commit y push del código del parcial a su repositorio. (commit)

5. (2 Puntos) Implemente un programa que reciba por línea de comandos dos números enteros. El primer número indica la cadencia en segundos a la cual se debe generar la señal SIGUSR1. El segundo parámetro corresponde al PID al cual se le enviará la SIGUSR1

Ejemplo:

- Cada 1 segundos envía SIGUSR1 al procesos con PID 2000  
`./2P_03.out 10 2000`