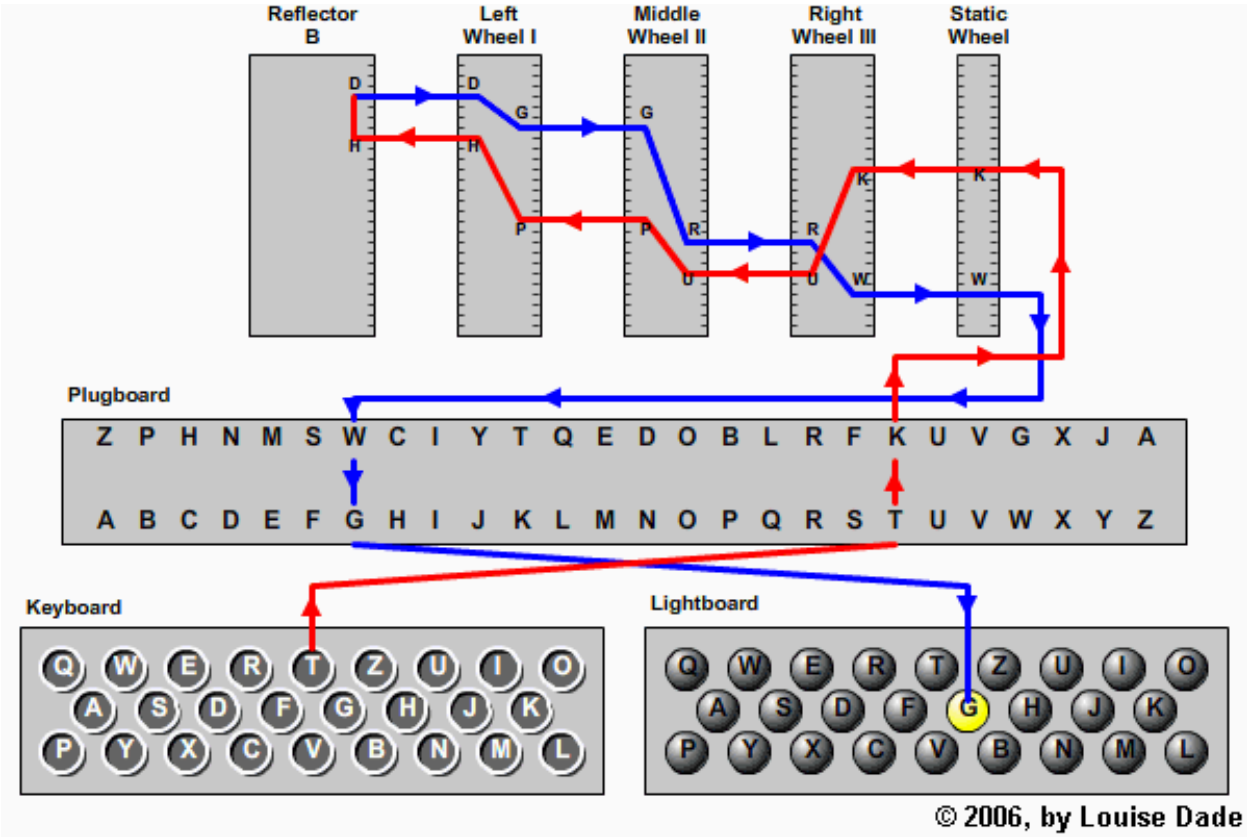
Tarea Programada #3

- La tarea debe entregarse al profesor a través del TEC Digital, el día y a la hora convenida.
- La tarea debe contener lo siguiente:
    a. Fuentes, todo el código necesario para ejecutar la tarea. El código debe estar debidamente documentado.
    b. Makefile para poder compilar los fuentes.
    c. Documentación, incluyendo al menos:
        i. Documentación en el código
        ii. Explicación del diseño y Arquitectura
- La tarea debe ser programada en YASM (Ensamblador) para Linux Ubuntu 12.X, en la sintaxis de Intel de x64.
- Toda tarea debe ser defendida ante el profesor, de tal manera todos los estudiantes deben poder explicar la solución satisfactoriamente.
- ¡Buena Suerte!

A Evaluar	**Puntos**	**Nota**
Documentación	10	
Implementación del Algoritmo de Encripción y Desencripción	60	
Lectura de los parámetros a través de los archivos	10	
Animación de la solución	10	
Completitud	10	
Total	**100**	
Lectura de los Rotores	5	
Total + Extra	**105**	

Enigma Machine

Durante la Segunda Guerra Mundial los alemanes usaron una maquina para encriptar y desencriptar los mensajes que se enviaban.
La tarea consiste en implementar una de estas máquinas usando ensamblador, esta usará el mismo procedimiento que en la segunda guerra mundial.
La maquina se compone de un plugboard que intercambia letras, así por ejemplo cuando se presiona la letra T, esta es intercambiada en el plugboard por la letra K.
Después, la letra pasa por tres rotores que que intercambian las letras, posteriormente un refector que intercambia la letra una vez más y regresa a travéz de los tres rotores, por el plugboard y posteriormente se enciende la letra encriptada.
Cada vez que se encripta una letra el rotor de la derecha avanza una posición, cuando el rotor derecho da una vuelta, se avanza el rotor del medio, y seguidamente cuando este da una vuelta se avanza el rotor de la izquierda.



Consideraciones:
1. Los rotores son fijos, serán los siguientes.

Rotor I -> `EKMFLGDQVZNTOWYHXUSPAIBRCJ`

Rotor II -> `AJDKSIRUXBLHWTMCQGZNPYFVOE`

Rotor III -> `BDFHJLCPRTXVZNYEIWGAKMUSQO`

Rotor IV -> `ESOVPZJAYQUIRHXLNFTGKDCMWB`

Rotor V -> `VZBRGITYUPSDNHLXAWMJQOFECK`

Reflector -> `JPGVOUMFYQBENHZRDKASXLICTW`

2. La configuración se carga de un archivo, este tendrá en primer lugar el orden de los rotores, en segundo lugar la posición inicial de cada rotor y por ultimo la configuración del plugboard.
Este es un ejemplo del archivo de configuración:

```
IV, V, I
23, 09, 20
XF, PZ, SQ, GR, AJ, UO, CN, BV, TM, Ki
```

3. Como segundo parámetro se escoje el archivo a encriptar o desencriptar y devuelve el resultado en el estándar out.

4. El proceso de encripción debe ser animado en la pantalla, donde se pueda ver el plugboard, los rotores y el reflector.

Puntos Extra

1. **Se dará valor extra si y solo si todos los otros puntos de la tarea funcionan bien.**
2. Implementar la lectura de los rotores por un archivo, donde cada línea sería el rotor, incluyendo el refector