

# Examen Laboratorio 2018

Fundamentos de Ordenadores y Sistemas Operativos

Mario Martínez Zarzuela  
marmar@tel.uva.es

# Enunciado

- Este segmento de datos está disponible en el archivo “data.asm”

```
1      .data
2  file:  .ascii  "Asimov.txt"
3  idfile: .word   -1
4  txterr: .ascii  "Error al abrir el archivo"
5  buffer: .space  1
6  fincad: .byte   '\0'
7  indexcar: .word  0,0,0,0,0,0,0,0 #Suponemos que archivo no tiene
8  ncarlin: .word  0,0,0,0,0,0,0,0
9  nlin:   .word   0
10 ncar:   .word   0
```

- Zonas de memoria:
  - **file**: nombre del archivo a leer.
  - **idfile**: servirá para almacenar el identificador de archivo.
  - **txterr**: mensaje de error.
  - **buffer**: espacio reservado en memoria.
  - **fincad**: caracter de '\0' para delimitar fin del buffer
  - **indexcar**: vector que almacenará el índice donde empiezan nuevas líneas en el archivo
  - **nlin**: almacenará el número de líneas del archivo
  - **ncar**: almacenará el número de caracteres del archivo
  - **ncarlin**: vector que almacenará el número de caracteres de cada línea del archivo

# NOTAS IMPORTANTES

- La **nota máxima** que se puede alcanzar es **2.30 puntos**
- Los archivos de código que **no compilen** o den un **error en ejecución** **obtendrán una puntuación de 0 puntos**
  - Por este motivo, se aconseja hacer cada apartado en un archivo diferente
- Utiliza **breakpoints** para para que sea más fácil depurar tu código

# Salida esperada del programa completo

- Apartado A

```
A robot may not injure a human being or, through inaction, allow a human being to come to harm.  
A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.  
A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.  
3  
-- program is finished running (dropped off bottom) --
```

- Apartado B

```
A robot may not injure a human being or, through inaction, allow a human being to come to harm.  
A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.  
A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.  
3 0 96 210 325
```

- Apartado C

```
A robot may not injure a human being or, through inaction, allow a human being to come to harm.  
A robot must obey the orders given it by human beings except where such orders would conflict with the First Law.  
A robot must protect its own existence as long as such protection does not conflict with the First or Second Laws.  
3 0 96 210 325  
  
96  
114  
115  
-- program is finished running (dropped off bottom) --
```

# Enunciado

- **A) (0.70)**
  - Escribe un programa en ensamblador que lea el archivo byte a byte en el espacio **buffer** de memoria
    - El identificador del archivo se debe almacenar en **idfile**
  - Si no se abre el archivo se debe mostrar el mensaje **txterr**

- Añade las instrucciones necesarias para ir imprimiendo el contenido del archivo en la consola según se va leyendo.
  - Para imprimir caracter a caracter en consola puedes usar la **llamada al sistema 11** y cargar el byte necesario en \$a0
- Al terminar la lectura deberá de almacenarse en **nlin** el número de líneas que tiene el archivo.
  - Para contar el número de líneas es posible contar el número de caracteres '\n'
  - Puedes añadir aquello que necesites en el segmento de datos.
- Añade las instrucciones necesarias para imprimir el número de líneas en la consola

- **B) (0.70)**

- Realiza las modificaciones oportunas para que el programa lleve por cuenta el número de caracteres del archivo y lo almacene en **ncar**.
- Cada vez que se encuentre una nueva línea, deberá almacenar en el vector **indexcar** el índice del caracter en el que empieza la línea
  - El primer elemento de **indexcar** deberá ser 0, puesto que es donde comienza la primera línea del archivo
  - Puedes comprobar la salida esperada en la diapositiva correspondiente de este enunciado



- Una vez leído el archivo, utiliza un bucle para escribir en consola los **nlin** elementos del vector **indexcar**, separados por un espacio
  - Puedes incluir en el segmento de datos aquello que creas necesario

- **C) (0.90)**

- Al final del programa anterior, añade las instrucciones necesarias para imprimir el número de caracteres que tiene cada línea del archivo
- Imprime cada valor en una nueva línea de la consola
  - Puedes comprobar la salida esperada en la diapositiva correspondiente de este enunciado
- Deberás calcular e imprimir los valores en un bucle, restando los valores almacenados en **indexcar**
- Además, deberás almacenar el resultado en **ncarlin**