



Proyecto 1 Septiembre – Diciembre 2023

Agenda estudiantil



1 Introducción

Debido a los problemas este trimestre para producir los comprobantes de inscripción, y en aras de volver a la presencialidad, DACE ha decidido distribuir a cada estudiante un pequeño dispositivo *pager*, el cual contendrá el horario y podrá ser actualizado remotamente. Usted ha sido seleccionado para programar la interfaz del dispositivo.

Debido a la naturaleza limitada del aparato, el programa debe escribirse en MIPS.

2 Requerimientos del programa

Diseñe y escriba un programa en MIPS capaz de almacenar y mostrar por texto un horario de clases. El horario debe almacenarse en un archivo `*.asm` leído usando la directiva `.include` descrita en el [manual de MARS](#). El programa debe funcionar aunque el archivo sea cambiado.

2.1 I/O

2.1.1 Interacción con el usuario

Para simular los botones del *pager*, se usarán las teclas WSAD donde

- W equivale a \uparrow , y debe mostrar la clase anterior en el horario
- S equivale a \downarrow , y debe mostrar la clase siguiente en el horario
- A equivale a \leftarrow , y debe mostrar el bloque horario equivalente en el día anterior
- D equivale a \rightarrow , y debe mostrar el bloque horario equivalente en el día siguiente

Si no hay una clase en el bloque equivalente al cambiar de día, debe mostrarse el horario como “libre”; sin embargo, al seleccionar una clase (W ó S), presionar el comando inverso (S ó W respectivamente) debe mostrar la clase siguiente en vez de volver a mostrar “Libre”.

Si se está visualizando la última clase del día, al introducir ‘S’, se debe mostrar la primera clase del día anterior, y vice-versa (si se está visualizando la primera clase del día, al introducir ‘W’, se debe mostrar la última clase del día anterior).

Si se está visualizando el último día de la semana, al introducir ‘D’, se debe mostrar el primero, y vice-versa (si se está visualizando el primer día de la semana, al introducir ‘A’, se debe mostrar el último)

Si se presiona una tecla dos veces seguidas, se debe avanzar ese número de elementos (así, si se está viendo el lunes a las 12:00, introducir dos ‘D’ debe mostrar el miércoles a las 12:00)

2.1.2 Formato de Salida

Debe imprimir una salida en un formato similar al siguiente

```

-----
| DDDD Hi - Hf                | ^
|                               | W
| COD-### NOMBRECURSO         | S
| EDF-### T/L                 | v
-----
<-A | D->

```

Donde

- DDDD es el nombre del día siendo visualizado (Lunes, Martes, etc.),
- H_i es la hora de inicio de la clase (donde 1 representa las 8 a.m., 2 representa las 9 a.m., etc.),
- H_f es la hora de finalización de la clase (no hay clases después de hora 12),
- COD-### es el código del curso en formato departamento-guión-número (el código siempre debe ocupar 7 caracteres),
- NOMBRECURSO es el nombre del curso,
- EDF-### es el aula del curso, también en 7 caracteres (o “TDD-000” si el curso es no-presencial),
- T/L es la frase “Teoría” o “Laboratorio” (si no aplica, pueden usarse frases aplicativas) y
- W, S, A y D aparecen simplemente como un recordatorio de la interfaz.

2.1.3 Formato de Archivo

El archivo con el horario debe tener el siguiente formato:

```

C1N: .asciiz "<COD-###1> <EDF-###1> <NOMBRECURSO1>"
C1H: .ascii "<T/L><H1iL><H1fL><T/L><H1iM><H1fM>...<T/L><H1iS><H1fS>"

C2N: .asciiz "<COD-###2> <NOMBRECURSO2>"
C2H: .ascii "<T/L><H2iL><H2fL><T/L><H2iM><H2fM>...<T/L><H2iS><H2fS>"

...

C8N: .asciiz "<COD-###2> <EDF-###1> <NOMBRECURSO2>"
C8H: .ascii "<T/L><H8iL><H8fL><T/L><H8iM><H8fM>...<T/L><H8iS><H8fS>"

```

Donde

- CjN es una etiqueta para el nombre del j -ésimo curso (COD-### siendo el código en 7 caracteres, EDF-### siendo el aula en 7 caracteres y NOMBRECURSO siendo el nombre),
- CjH es una etiqueta para el horario del j -ésimo curso (*ibidem*),
- Hj_{iD} es la hora de inicio del j -ésimo curso el día D (L=Lunes, M=Martes, ... S=Sábado) en dos dígitos (donde 01 representa las 8 a.m., 02 representa las 9 a.m., etc.) y
- Hj_{fD} es la hora de finalización del j -ésimo curso el día D .

Si una clase no tiene horario en un día de la semana específico, se debe usar la hora 00

Todos los archivos de horario deben tener exactamente 8 entradas. Si el horario tiene menos cursos, las etiquetas CjN correspondientes deben apuntar al string "-----" (nueve guiones).

2.1.4 Ejemplo de archivo de entrada

Un archivo de entrada para un estudiante *pensum* en el trimestre en curso podría ser

```
C1N: .asciiz "CI-2613 MEM-008 Algoritmos III"
C1H: .ascii  "-0000T0304-0000T0304-0000-0000"

C2N: .asciiz "CI-2693 MYS-019 Lab. Algoritmos III"
C2H: .ascii  "-0000-0000L0204-0000-0000-0000"

C3N: .asciiz "CI-3815 TDD-000 Organizacion del Computador"
C3H: .ascii  "T0607-0000T0607-0000L0607-0000"

C4N: .asciiz "CO-3121 MEM-005 Probabilidad"
C4H: .ascii  "T0304-0000T0304-0000-0000-0000"

C5N: .asciiz "FLX-418 AUL-011 El Razonamiento Logico"
C5H: .ascii  "-0000-0000-0000T0204-0000-0000"

C6N: .asciiz "-----"
C6H: .ascii  "-0000-0000-0000-0000-0000-0000"

C7N: .asciiz "-----"
C7H: .ascii  "-0000-0000-0000-0000-0000-0000"

C8N: .asciiz "-----"
C8H: .ascii  "-0000-0000-0000-0000-0000-0000"
```

Donde FLX-418 es un Estudio General, y – en CjH indica que no es ni teoría ni laboratorio.

2.1.5 Corrida de ejemplo

```
-----
|Lunes 01 - 01                                | ^
|                                                | W
|          Libre                               | S
|                                                | v
-----
<-A | D->
```

S

```
-----  
|Lunes 03 - 04                                | ^  
|                                                | W  
|CO-3121 Probabilidad                          | S  
|MEM-005 Teoria                               | v  
-----  
<-A | D->
```

D

```
-----  
|Martes 03 - 04                                | ^  
|                                                | W  
|CI-2613 Algoritmos III                       | S  
|MEM-008 Teoria                               | v  
-----  
<-A | D->
```

D

```
-----  
|Miercoles 03 - 04                             | ^  
|                                                | W  
|CO-3121 Probabilidad                          | S  
|MEM-005 Teoria                               | v  
-----  
<-A | D->
```

S

```
-----  
|Miercoles 06 - 07                             | ^  
|                                                | W  
|CI-3815 Organizacion del Computador          | S  
|TDD-000 Teoria                               | v  
-----  
<-A | D->
```

S

```
-----  
|Jueves 02 - 04                                | ^  
|                                                | W  
|FLX-418 El Razonamiento Logico               | S  
|AUL-011 Teoria                               | v  
-----  
<-A | D->
```

2.2 Organización del programa.

Debe diseñar una forma de cargar el horario que incluya listas enlazadas (en particular, utilicen las ventajas de la lista enlazada para saltar al siguiente curso en vez de perder tiempo de procesamiento revisando todas las horas subsiguientes).

Debe incluir un archivo de entrada por cada miembro del grupo con horarios diferentes. Al menos uno de estos horarios debe incluir el curso “Organización del Computador”. El programa debe ser capaz de imprimir los nombres de los cursos incluidos (observe que “Organización del Computador” tiene 27 caracteres).

3 Informe

Debe entregar un informe donde se detallen las decisiones tomadas en el proyecto incluyendo:

- Introducción, dando la motivación y estructura del informe
- Diseño de la estructura de datos para almacenar el horario
- Dificultades Encontradas en la implementación u otras decisiones posteriores
- Conclusiones, dando el estado actual del programa y las lecciones aprendidas

4 Evaluación

El proyecto tiene una ponderación de 10 puntos. Se asignarán

- 5 puntos por código
 - 1 punto por su estructura de datos
 - 1 punto por la(s) función(es) de acceso a la estructura de datos
 - 1 punto por su función para procesar el archivo de entrada
 - 1 punto por la impresión de la salida
 - 1 punto por su programa principal
- 2 puntos por ejecución
 - 0,5 puntos por mostrar correctamente el elemento anterior al presionar W
 - 0,5 puntos por mostrar correctamente el elemento siguiente al presionar S
 - 0,5 puntos por mostrar correctamente el elemento correspondiente en el día anterior al presionar A
 - 0,5 puntos por mostrar correctamente el elemento correspondiente en el día siguiente al presionar D
- 3 puntos por su informe
 - 0,5 puntos por su introducción
 - 1 punto por el diseño de la estructura de datos
 - 1 punto por las dificultades encontradas
 - 0,5 puntos por sus conclusiones

Todo su código debe estar debidamente documentado y siguiendo las convenciones de MIPS.

El programa debe correr sin errores.

5 Extra Credit

- Simule el botón de apagado (línea verde en la foto de la página 1), permitiendo que introducir 'X' termine el programa
- Agregar soporte para otros tipos de horario (distintos a teoría y laboratorio)
- Investigue el Syscall 30 en el [manual de MARS](#) y utilícelo para comenzar el programa en la clase correspondiente al día y hora retornadas.
- Investigue el Syscall 31 y utilícelo para dar una alerta cuando se acerque la hora de la siguiente clase.
- Simule el botón de "mute" (círculo rojo en la foto de la página 1) permitiendo que, al introducir 'M', se silencie la alerta o se vuelva a activar. La alerta debe mostrar un texto en pantalla si está silenciado.

Para que cualquiera de estas acciones extra sea considerada, debe ser reportada en el informe.