

Proyecto 1 (16%)

Estructuras de datos y Subrutinas

Objetivo

Familiarizarse con la representación de datos compuestos y programación estructurada en el ambiente de programación *MARS* y con el lenguaje ensamblador *MIPS*.

Objetivos Específicos

1. Adquirir destrezas en el lenguaje de bajo nivel del simulador *MARS*
2. Adquirir destrezas en el manejo eficiente de los recursos (registros y memoria).
3. Adquirir destrezas en la definición de estructuras complejas en un ambiente de programación de ensamblador.
4. Adquirir destrezas en el desarrollo de aplicaciones que usen subrutinas en el lenguaje de programación *MIPS*.

Enunciado

En este proyecto Ud deberá diseñar e implementar un prototipo de un sistema de asignación de cupos para dar soporte a proceso de inscripción de materias dictadas por una institución ficticia de educación superior. Este proceso involucra las siguientes actividades:

1. Los estudiantes realizan la solicitud de inscripción de un conjunto de materias.
2. El coordinador recibe un archivo de texto con estas solicitudes.
3. **El coordinador utiliza su programa para procesar las solicitudes.**
4. El coordinador publica un archivo de texto con la **lista tentativa de inscritos** por materia y el número de cupos libres.
5. Se realiza el proceso de solicitud de corrección de inscripción.
6. El coordinador recibe las solicitudes de corrección de los estudiantes en otro archivo de texto.
7. **El coordinador utiliza su programa para procesar este segundo conjunto de solicitudes.**

8. El coordinador publica la **lista definitiva de inscritos** por materia en un archivo de texto.

Note que su prototipo sólo asiste al coordinador en los puntos 3 y 7. Estos se realizan en secuencia dentro del mismo programa. En otras palabras, Ud no debe escribir dos programas diferentes. Ud debe escribir un solo programa que:

- primero procese un archivo con las solicitudes de inscripción
- y luego procese el otro que contiene las solicitudes de corrección de inscripción.

Formato de los Archivos

Se presentan los siguientes formatos.

Archivo de Estudiantes

Tendrá una línea por estudiante con la siguiente información:

- **Carnet:** 8 caracteres. Tamaño fijo.
- **Nombre:** Máximo 20 caracteres. Estará entre comillas dobles.
- **Índice:** 6 Caracteres. Redondeado a 4 decimales. Tamaño fijo.
- **Créditos aprobados:** 3 caracteres. Tamaño fijo.

Archivo de Materias

Tendrá una línea por materia con la siguiente información:

- **Código:** 7 caracteres. Tamaño fijo.
- **Nombre:** Máximo 30 caracteres. Estará entre comillas dobles.
- **Número de créditos:** 1 caracter. Tamaño fijo
- **Cupo:** 3 caracteres. Tamaño fijo.
- **Número Mínimo de créditos aprobados:** 3 caracteres. Tamaño fijo

Archivo de Solicitudes de Inscripción

Tendrá una línea por cada par Estudiante, Materia:

- **Carnet del Estudiante:** 8 caracteres. Tamaño fijo.
- **Código Materia:** 7 caracteres. Tamaño fijo

Archivo de Solicitudes de Corrección

Tendrá una línea por cada terna Estudiante, Materia, Operación:

- **Carnet del Estudiante:** 8 caracteres. Tamaño fijo.
- **Código Materia:** 7 caracteres. Tamaño fijo
- **Operación:** 1 carácter. Valores posibles: I o E (mayúsculas), Inclusión y Eliminación respectivamente.

Archivo Con Inscritos tentativos y definitivos

Estos dos archivos tendrá un formato más complejo que los anteriores. Presentará los inscritos agrupándolos por materia:

- Primera línea de cada materia:
 - **Código Materia:** 7 caracteres.
 - **Nombre:** Máximo 30 caracteres.
 - **Número de Cupos libres:** 3 caracteres.
- Resto de las líneas (los estudiantes inscritos):
 - **Carnet del Estudiante:**
 - **Nombre:** Máximo 20 caracteres.
 - **Tipo Corrección:** sólo para las correcciones. Indica si:
 - Inclusión en Corrección (I)
 - Eliminación en Corrección (E)

Debe haber una línea en blanco de separación entre materias. Y un espacio en blanco de separación entre los campos.

Para cada uno de estos formatos, se presenta un ejemplo.

Fases

Ud debe realizar las siguientes tareas:

1. Definir estructuras de datos apropiadas para el problema planteado
2. Cargar los datos contenidos en los archivos de estudiantes, materias y solicitudes de inscripción.
3. Procesar las solicitudes de inscripción en base a un conjunto de criterios. Corresponde con la **Actividad 3** descrita antes. Debe generar el archivo con la **lista tentativa de inscripciones**.

4. Procesar las solicitudes de correcciones en base a un conjunto de criterios. Corresponde con la **Actividad 7** descrita antes. Debe generar el archivo con la lista definitiva de inscripciones.

Criterios

Criterios

Se presenta dos modalidades para procesar las eliminaciones e inclusiones, en este orden.

Modalidad 1:

En el proceso de solicitud de inscripción: se aceptan todas las solicitudes. Este criterio no considera límite de cupos.

En el proceso de corrección:

- En el caso de las eliminaciones:
 - Todas deben ser aceptadas.
- En el caso de las inclusiones, se debe tener en cuenta uno de estos criterios:
 1. Se le da prioridad a los estudiantes de mayor índice.
 2. Se le da prioridad a los estudiantes que tengan un mayor número de materias inscritas.
 3. Se le da prioridad a los estudiantes de cohortes más recientes.
 4. Se le da prioridad a los estudiantes con menor número de créditos inscritos.
 5. Se le da prioridad a los estudiantes con menor número de créditos aprobados.

Modalidad 2:

En el proceso de solicitud de inscripción: **no** se aceptan todas las solicitudes, se considera límite de cupo. Se aceptan solicitudes mientras haya cupos libres, considerando el orden de “llegada”.

En el proceso de corrección:

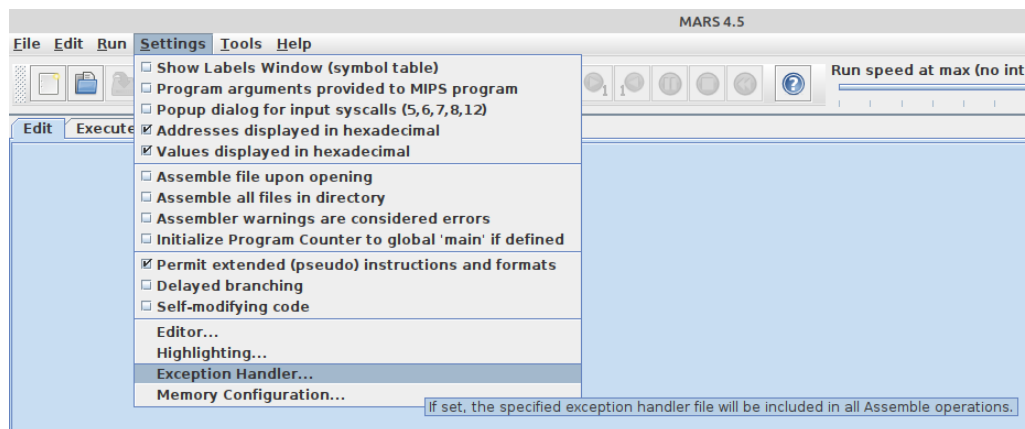
- En el caso de las eliminaciones:
 - Todas deben ser aceptadas.
- En el caso de las inclusiones, se debe tener en cuenta uno de estos criterios:
 1. Se le da prioridad a los estudiantes de menor índice.

2. Se le da prioridad a los estudiantes que tengan un menor número de materias inscritas.
3. Se le da prioridad a los estudiantes de cohortes menos recientes.
4. Se le da prioridad a los estudiantes con mayor número de créditos inscritos.
5. Se le da prioridad a los estudiantes con mayor número de créditos aprobados.

En semana 5: a cada equipo se le asignará un criterio diferente. Envíe un correo a su profesor indicando los integrantes de su equipo para realizar esta asignación.

Nota: Al final del proceso, el número de estudiantes efectivamente inscritos por materia debe no debe exceder el cupo establecido.

Los nombres de los archivos estarán en el archivo [exceptions.s](#) que Ud debe usar como *archivo Exception Handler*. Ver opción *Settings* en el menú *Simulator* de *MARS*. Verifique que sus simulador esté configurado de esta forma.



Puede descargar un ejemplo del archivo myexception.s.

NOTAS IMPORTANTES:

1. Observe el archivo *myexceptions.s* provisto, genere casos de pruebas suficientes para verificar que su programa funciona correctamente. **No** pruebe su programa solamente con ese ejemplo, la corrida del programa se realizará con varios casos de pruebas generados por el equipo profesoral.
2. No seguir las especificaciones aquí dadas podría resultar en la no corrección de su tarea.
3. Si tiene preguntas, se le recomienda aclararlas con tiempo.

Entrega

La tarea deberá ser entregada el **día jueves de la semana (7) hasta las 11:59am** al correo electrónico eduardo.AT.usb.ve y debe adjuntar un único archivo con nombre **Proy-Apellido1-Apellido2.tar** que contenga **un directorio/carpeta con el nombre Proy1-Apellido1-Apellido2**. **Éste debe contener:**

1. Los fuentes en formato .s:

Estos archivos serán la implementación de su proyecto y se usará *para ejecutarlo*.

2. Los fuentes en formato .pdf:

Estos archivos serán la implementación de su proyecto y será usado para *evaluar su código*.

3. Archivo Informe.pdf:

Debe contener un informe (5 páginas máximo) que explique su diseño y las consideraciones tomadas en cuenta en su implementación.

4. Los conjuntos de archivos en los puntos 1 y 2 deben corresponder con la misma versión de su proyecto. Alguna discrepancia puede resultar en la **no corrección de su proyecto.**

5. Su código debe estar debidamente documentado y estructurado.

Tome en cuenta que:

- El código del programa deberá estar claro, conciso y bien documentado. Deberá tener en comentario un encabezado con el nombre y carnet del autor, así como la planificación de registros en las secciones que lo ameriten.
- En la documentación del código NO se debe comentar cada una de las líneas. Se deben documentar aquellas secciones que sean de interés.

Enlaces de interés:

- http://courses.missouristate.edu/kenvollmar/mars/Help/Help_4_1/SyscallHelp.html

EB/EB