

Proyecto de Autómatas y Lenguajes

Curso 18/19

Grupo Miércoles 16:00-18:00

Objetivo de la sesión

Esta sesión tiene los siguientes objetivos

- **Procedural**
 - Comprender el uso de olib.o
 - Comprender las diferencias entre nasm y otros ensambladores
 - Comenzar a estudiar el nasm que usaremos este año.
 - Avanzar (aprox. 1/2) práctica NASM
- **OO**
 - Decisiones de diseño del grafo de la tabla de símbolos
 - Anotación de funcionalidad básica

Plan de trabajo

OO1 (común) lol.ol	OO2 doc ts	OO3 (común) doc ts	OO4 (común) recapitulación TS	OO5 trabajo personal
P1 análisis nasm	P2 (común) análisis nasm	P3 escritura nasm	P4 plantilla común	P5 (común) plantilla común

Material de consulta

- Este guión
- **Procedural**
 - El material de Moodle común relacionado con
 - La introducción a nasm
 - La descripción de olib.o
 - El manual de NASM
- **OO**
 - El material introductorio a la estructura de la tabla de símbolos de o

Actividad P1: PROCEDURAL

INTRODUCCIÓN A NASM

Análisis de programas nasm (trabajo alumnos) (20 min)

- Las parejas de procedural se separan en 2 (1s y 2s)
- Formación de grupos (tríos) de 1s y 2s.
- Análisis de programas de ejemplo
- Debéis escribir en una hoja los nombres de los miembros del trío, se pondrá en común en clase (posiblemente preguntando a los tríos).

1. analisis1.asm <ul style="list-style-type: none"> ● Guarda y restauración de puntero de pila ● Estructura general del programa ● Declaración de variables con valor ● Declaración de variables sin valor ● Convenio de llamadas a funciones C ● Ejemplo de uso de función de alfabib.o: <ul style="list-style-type: none"> ○ Preparación del código para llamar funciones externas ○ print_int ● Cómo compilar y ejecutar código nasm 	resumen nasm t2-5 resumen nasm t2-5 resumen nasm t10 resumen nasm t7-9 librería olib t2-3 resumen nasm t17
2. analisis2.asm <ul style="list-style-type: none"> ● Uso de especificador de tamaño dword ● Asignar valor a una variable ● Asignar valor a un registro ● Asignar el valor de un registro a una variable ● Introducir en la pila una constante ● Introducir en la pila una variable ● Introducir en la pila un registro ● Extraer valores de la pila 	resumen nasm t11-12
3. analisis3.asm <ul style="list-style-type: none"> ● Realizar una operación aritmética con la decisión de gestión de memoria auxiliar de este curso: <ul style="list-style-type: none"> ○ Uso de pila para espacio auxiliar ○ Peculiaridades de div, imul 	resumen nasm t13-16
4. analisis4.asm <ul style="list-style-type: none"> ● Familiarizarse con algunos datos de tipo boolean 	resumen libreria olib
5. analisis5.asm <ul style="list-style-type: none"> ● Cómo escribir textos con mensajes 	resumen libreria olib

Actividad OO1: OO (20 min)

Esta actividad se realizará por grupos de compilador de la parte OO, sería conveniente que el elemento “mixto” también la realizará

- Toma como punto de partida el programa lol.ol de la pasada sesión (refréscalo ya que la versión actual ya está corregida)
- Discutamos qué debería ocurrir en la tabla de símbolos a medida que compilamos el programa
 - Esta actividad se hará mediante interacción con el profesor

Actividad OO2: OO (15 min)

Con la misma distribución de estudiantes anterior.

Consulta el material sobre la estructura de la tabla de símbolos y busca respuestas a las siguientes preguntas que posteriormente se pondrán en común con tu profesor y el resto de la clase

¿Debe ser el grafo dirigido o no (bidireccional)?	t4
¿Cómo accederás a los nodos de las clases?	t6
¿A qué nos referiremos por <code>tablaAmbitos</code> ?	t7...
¿A qué clase pertenecerá el ámbito main?	t11
¿Qué entenderemos por <code>tablaSimbolos</code> ?	t13
¿Cuántas tablas de símbolos tiene que tener al menos una tabla de ámbitos?	t16
¿Qué entenderemos por <code>tablaSimbolosClaes</code> ?	t18

Actividad P2: PROCEDURAL (15 min)

Se trata de poner en común lo trabajado en la actividad P1.

Actividad P3: PROCEDURAL (20 min)

Con la misma distribución de alumnos de P1 y P2

Cada trío escribirá alguno de los siguientes programas (se recomienda que cada miembro del trío escriba uno)

1. generado1.asm <pre>int main() { int x = 8; printf x; }</pre>	Declaración de variable inicializada Escritura de variable
2. generado2.asm <pre>int main() { int x; x = 8; printf x; }</pre>	Declaración de variable sin inicializar Asignación de valor a variable Escritura de variable
3. generado3.asm <pre>int main() { int x; int y = 10;</pre>	Declaración de variables inicializadas y no Lectura de variable Expresión aritmética

<pre>scanf x; x = x * 2 -y; printf x; }</pre>	Asignación Escritura de variable
--	-------------------------------------

Sube a la correspondiente tarea de Moodle un zip con los ficheros que hayas podido completar... por favor, sólo aquellos que compilen y funcionen.

Actividad OO3: OO (20 min)

Con la misma distribución de alumnos de OO2
Puesta en común de lo trabajado en OO2.

Actividad P4: PROCEDURAL (20 min)

Se vuelve a los grupos de compilador

- Accede al documento de drive sobre la plantilla común de generacion
- Seleccionar cada grupo (comenzando con el 1) tantas funciones como le corresponda rotando
 - g1 -> t1
 - g2 -> t2
 - g3 -> t3
 - ...
 - gn-> tn
 - g1 -> tn+1 etc...
- Escribe el nombre del grupo en la cabecera de la función
- Escribe el pseudocódigo de la función que te toca.

Actividad P5: PROCEDURAL (hasta el final de la sesión o para la próxima)

Puesta en común de la plantilla común.

Actividad OO4: OO (algunos minutos)

Con la misma distribución de alumnos de OO2

Recapitulación TS

- Estructura
- Tipo de datos de los nodos
- Tipo de dato de las tablas hash
- Cómo se identifica cada elemento?
- Funcionalidad básica

Actividad OO5: OO (algunos minutos)

Trabajo personal.

PARA LA PRÓXIMA SEMANA

OO

- Deberían tener casi acabado el grafo con la funcionalidad estudiada

PP

- Sólo deberíamos estar pendientes de las funciones que hacen comparaciones o la negación (aquellas que usan etiquetas)