

1151073 - Programación de Sistemas
Trimestre: 2017-I Grupo: CSI81

Evaluación Terminal

Sección 1 Puntaje máximo: 100

1. (5 puntos) Es un programa que toma código en lenguaje ensamblador y lo traduce a código máquina: _____.
2. (5 puntos) Es un programa que se toma uno o más códigos objeto y calcula las direcciones finales de las secciones y etiquetas para producir el ejecutable final: _____.
3. (5 puntos) Es un programa que toma el código ejecutable de un programa, lo carga en memoria e inicia su ejecución: _____.

Dado el siguiente código fuente en ensamblador:

```
EXPORT eti

SECTION text
    INC RX0
eti: JMP eti
    MOV RX1, v1

SECTION data
    ALLOC WORD v1, -129
```

4. (15 puntos) Muestre el contenido de todas las secciones y la tabla de símbolos suponiendo que cada sección individual comienza en la dirección 0.
5. (30 puntos) Muestre el contenido de todas las secciones y la tabla de símbolos con las direcciones finales de cada etiqueta, suponiendo que las direcciones de inicio de las secciones serán: `text (10)`, `data (20)`.
6. (40 puntos) Genere un código objeto para el código fuente en ensamblador, suponiendo que todas las secciones comienzan en la dirección cero. Incluya en la tabla de símbolos al menos las etiquetas exportadas.

Sección 2

Puntaje máximo: 100

1. (10 puntos) Menciona una ventaja y una desventaja del enlazado estático.
2. (10 puntos) Menciona una ventaja y una desventaja del enlazado dinámico.
3. (5 puntos) Genera el código máquina de la siguiente sección `text`, suponiendo inicia a partir de: (a) la dirección 0, (b) la dirección 10.

```
SECTION text
    JMP eti
    NOP
eti: NOP
```

4. (15 puntos) Traduzca el código anterior a una versión independiente de su posición y genere también el código máquina suponiendo que la sección `text` inicia a partir de: (a) la dirección 0, (b) la dirección 10.
5. (10 puntos) Dada la siguiente tabla de mapeos entre la memoria física y virtual de un proceso, calcule:
 - a. La dirección física de la dirección virtual 15.
 - b. La dirección física de la dirección virtual 16.
 - c. La dirección virtual de la dirección física 8.

Direcciones virtuales	Direcciones física
De 0 a 15	De 1024 a 1039
De 16 a 31	De 0 a 15

6. (25 puntos) Desensamble (genere el código ensamblador equivalente de) los códigos objeto mostrados al final de esta sección del examen.
7. (25 puntos) Calcule el ejecutable final que se produce si el enlazador toma los códigos objeto mencionados y `main` es el punto de entrada.

Sección 3

Puntaje máximo: 100

1. (20 puntos) Dado el siguiente código fuente en ensamblador, escriba el código resultante de la expansión de macros.

```
BEGIN MACRO examen
eti: NOP
    JMP eti
END MACRO
```

```
SECTION text
    MOV RX0, -5
    EXPAND examen
    MOV RX0, 0
    EXPAND examen
    MOV RX0, +5
```

2. (20 puntos) Dado el siguiente código fuente en C++, escriba el código resultante de la expansión y evaluación de macros.

```
#define intercambia(a, b) \
    { int c = a; a = b; b = c; }

int main( )
{
    int v1 = 12;
    int v2 = 34;
    intercambia(v1, v2);

    int raro = __LINE__;

    int v3 = 56;
    int v4 = 78;
    intercambia(++v3, ++v4);
}
```

3. (30 puntos) Ejecute el siguiente código en C++:

```
#include <algorithm>
#include <cstdio>
#include <cstdlib>
#include <functional>

void procesa(int* ini, int* fin)
{
    int* p = std::min_element(ini, fin);
    std::transform(ini, fin, ini,
std::bind2nd(std::minus<int>( ), *p));
```

```

    for (int* i = ini; i != fin; ++i) {
        *i /= *p;
    }
}

int main( )
{
    int arr[16];
    for (int i = 0; i < 16; ++i) {
        arr[i] = std::rand( );
    }

    procesa(arr, arr + 16);

    for (int i = 0; i < 16; ++i) {
        std::printf("%d", arr[i]);
    }
}

```

En caso de que se interrumpa la ejecución del programa, use un depurador para encontrar la causa. Imprima la traza de llamadas a función e imprima todos los valores de las variables locales del marco en el que ocurre la interrupción del programa. En caso de que no se interrumpa el programa, imprima la secuencia de los valores iniciales del arreglo y el valor de todas las variables accesibles desde la función `procesa` antes y después del ciclo `for`.

4. (15 puntos) Describa la diferencia entre un compilador y un intérprete.
5. (15 puntos) El proceso de identificar cuál es la acción de la siguiente instrucción a ejecutar es una de las tareas más costosas en un intérprete. Si el tipo de una instrucción se identifica mediante un entero, describa las ventajas y desventajas de las siguientes técnicas de implementación: sentencias `if-else` encadenadas, sentencia `switch`, hilado directo.