



Arquitecturas Gráficas

Proyecto 1



Universidad
Rey Juan Carlos

1. Objetivo

El alumno será capaz de programar sentencias de control, bucles y estructuras de datos sencillas en ensamblador. Para ello, se realizará la traducción de varios códigos C a ensamblador. De esta forma, habrá adquirido los conocimientos básicos del funcionamiento de un computador así como los fundamentos del lenguaje ensamblador necesarios para poder desarrollar programas más complejos.

2. Condiciones de entrega

Cómo.

- Todos los ficheros que se soliciten deben subirse agrupados en un fichero comprimido.
- El fichero debe tener el siguiente nombre:

AG_grupo_Apellido1_Apellido2.zip

Por ejemplo, un equipo del grupo 2, formado por Luis Aranda y Adrián López, debería tener el nombre *AG_2_Aranda_Lopez.zip*.

- Si dentro falta alguno de los ficheros solicitados, se considerará como una práctica incompleta y por tanto suspensa.

Cuándo.

- La fecha límite se indica en la actividad habilitada en el Aula Virtual de la asignatura.

Qué.

- Tres programas en ensamblador (.asm) con los códigos de los ejercicios propuestos en la práctica. Se indicará al principio de cada fichero, con comentarios, el ejercicio al que hace referencia y el nombre de los alumnos autores de dicho código.

Prohibiciones.

- NO se admiten entregas por medios que no sea la actividad habilitada en el Aula Virtual.
- NO se admiten entregas fuera del plazo marcado.
- Códigos copiados implican nota de 0 para el copiado y para el copiador.
- Los códigos tienen que ser originales de los miembros del grupo. Si otra persona ajena al grupo los realizara se desencadenarán los procedimientos sancionadores correspondientes.

3. Lista de comprobación antes de que termine el plazo

Esta lista te sirve para repasar que cumples con todos los requisitos formales de la práctica.

- ¿Te has apuntado en un grupo a través de la actividad del aula virtual?
- ¿Has metido todo lo necesario para evaluar la práctica en el fichero comprimido?
- ¿Tienen todos esos archivos el formato solicitado?
- ¿Has nombrado el fichero comprimido con el patrón indicado arriba?
- ¿Has subido el fichero comprimido?

Si quieres, basta con que subas la práctica. No importa que se quede como borrador porque de todos modos queda la fecha. La ventaja es que, si te has equivocado en algo, estando como borrador puedes subir tantas veces como quieras el fichero.

4. Enunciado

Esta práctica consta de 3 ejercicios a realizar. Con cada ejercicio se trabajan unos aspectos concretos de ensamblador vistos durante las sesiones de prácticas en clase. El procedimiento es el mismo en cada ejercicio, dado un código en lenguaje C, el grupo debe realizar una traducción a ensamblador lo más fiel posible. Intentando mantener nombres de variables, reservas de memoria, bucles, estructuras de datos, etc.

1. Ejercicio 1 – Direccionamiento y condiciones if-else

```
int main()
{
    int A=15; // Cambiar los valores de A, B y C para comprobar
    int B=10; // que entra en las distintas condiciones if-else
    int C=5;
    int D=2;
    int Z=0;

    if((A > B) || ((C+1) == 7)){
        Z = Z - 3;
    }
    else if((A < B) && (C > 5)){
        Z = 2;
    }
    else{
        Z = (A-B) * (C+D) - (A/C);
    }
    return 0;
}
```

2. Ejercicio 2 – Vectores y bucles

```
int main()
{
    int A[5]; // Región de memoria vacía para almacenar 5 elementos
    int B[6] = {0,1,2,3,4,5};
    int i;

    for(i=0; i<5; i++) {
        A[i] = B[i] + B[i+1];
    }

    i--;
    while(i >= 0) {
        A[i] = A[i] * 2;
        i--;
    }
    return 0;
}
```

3. Ejercicio 3 – Cadenas de caracteres y entrada/salida

```
int main()
{
    char cadena[10];
    int i=0;
    char *result = NULL; // En binario, NULL es cero

    // Leemos una cadena introducida por el usuario
    printf("Introduzca una cadena: ");
    scanf("%s", cadena);

    // Buscamos en la cadena la letra 'm'.
    // El resultado será un puntero a la primera m que haya
    // o NULL si no hay
    while(cadena[i] != '\0') {
        if(cadena[i] == 'm') {
            result = &cadena[i];
            break; // Dejamos de buscar al encontrar la primera m
        }
        i++;
    }

    if(result != NULL)
        printf("La primera m está en la dirección %d\n", result);
    else
        printf("La cadena no contiene la letra m\n");

    return 0;
}
```

Puntuación de cada ejercicio:

- (5 puntos) El programa ensambla y funciona correctamente.
- (2 puntos) El programa en ensamblador es fiel al programa en C.
- (2 puntos) Los registros y la memoria de datos del procesador se utilizan y gestionan correctamente.
- (1 punto) El código está comentado y se entiende perfectamente.

La **nota final** del proyecto 1 se calculará realizando la media de los tres ejercicios.