COGNOMS:	. NOM:
1er Control Arquitectura de Computadors	Curs 2016-2017 Q1

- Temps: 13:30 a 15:00
- Poseu clarament amb LLETRES MAJÚSCULES a cada full els cognoms i el nom

## Problema 1. (3 puntos)

Dado el siguiente código escrito en C, que compilamos para un sistema linux de 32 bits:

```
typedef struct {
  char c;
  char d;
  short u;
  short e[2];
  char f[4];
  float *g;
  int h;
  short i;
} s1;
```

a) **Dibuja** como quedarían almacenadas en memoria las estructuras **s1** y **s2**, indicando claramente los desplazamientos respecto al inicio, el tamaño de todos los campos y el tamaño de los structs.

```
c <- 0
   ---- d <- 1
                          u
  e[0] | d | c | e <- 2
----- <- 4
                          v[0]
                                |f[1]|f[0]| e[1] | f < -6
  ----- <- 8
 ---- |f[3]|f[2]|
 ----- <- 12
                         v[99]
                          ----- <- 2404
     *g
           ----- <- 16
                    | --- | w |
                     ----- <- 2408
     h
           ---- | i |
                    Tamaño s2: 2408 bytes
Tamaño s1: 24 bytes
```

b) **Escribe** UNA ÚNICA INSTRUCCIÓN que permita mover **x.v[10].d** al registro **%al**, siendo **x** una variable de tipo **s2** cuya dirección está almacenada en el registro **%ecx**. Indica la expresión aritmética utilizada para el cálculo de la dirección

```
La expresión aritmética para calcular la dirección del operando es: @x+4+24*10+1, por lo tanto %ecx+245 movb 245(%ecx), %al
```

c) Escribe UN CONJUNTO DE 2 INSTRUCCIONES que permita mover x.v[y.h].f[2] al registro %al, siendo x una variable de tipo s2 cuya dirección está almacenada en el registro %ecx e y una variable de tipo s1 cuya dirección está almacenada en el registro %ebx. Indica la expresión aritmética utilizada para el cálculo de la dirección.

```
La expresión aritmética para calcular la dirección del operando es: @x+4+M[@y+16]*24+8 imul $24, 16(%ebx), %eax movb 12(%ecx, %eax), %al
```

5 April 2018 12:11 pm 1/4

Poseu clarament amb LLETRES MAJÚSCULES a cada full els cognoms i el nom

## Problema 2. (3 puntos)

Dado el siguiente código escrito en C:

```
int examen(int a[5], int b[5], int x) {
  int c[5], y;
  y=1;
  ...
}
```

 a) Dibuja el bloque de activación de la rutina examen, indicando claramente los desplazamientos respecto a %ebp y el tamaño de todos los campos.

```
examen
                                           20
                                                            -24
                                                    c[]
                                            4
                                                            -4
                                                    У
                                            4
                                                 ebp viejo
                                                            <- %ebp
                                                dir. retorno
                                            4
                                            4
                                                            +8
                                                   @a
                                            4
                                                            +12
                                                   @b
                                            4
                                                            +16
                                                    Х
```

b) **Traduce** a ensamblador x86 el inicio de la rutina examen (tres primeras lineas en C hasta la sentencia y=1;) sabiendo que usa los registros **%eax**, **%ebx**, **%ecx** y **%edx**.

```
examen: pushl %ebp

movl %esp, %ebp

subl $24, %esp;

pushl %ebx

movl $1, -4(%ebp)
```

c) **Traduce** a ensamblador x86 la siguiente sentencia en C que se encuentra en el cuerpo de la rutina examen.

```
y = examen(b, c, a[3]);
```

```
movl 8(%ebp), %ebx
pushl 12(%ebx)
leal -24(%ebp), %eax
pushl %eax
pushl 12(%ebp)
call examen
addl $12, %esp
movl %eax, -4(%ebp)
```

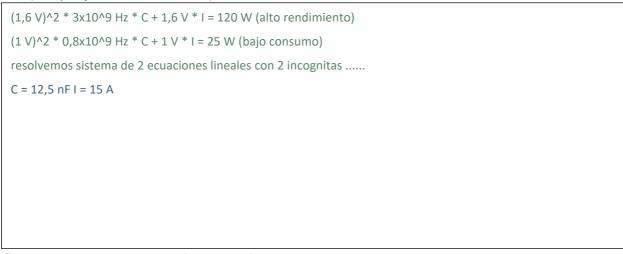
5 April 2018 12:11 pm 2/4

COGNOMS:	NOM:
1er Control Arquitectura de Computadors	Curs 2016-2017 Q1
Poseu clarament amb LLETRES MAJÚSCULES a cada full els cognoms i el	nom
Problema 3. (4 puntos)	
En un ordenador en el que tenemos instalado el entorno usado en el labor programa de 1000 instrucciones ensamblador se ha ejecutado en 2 segund $4,8x10^9$ instrucciones y $2,4x10^8$ operaciones de coma flotante que debido a la 10 instrucciones cada una.	los usando 6x10 <sup>9</sup> ciclos y ha ejecutado
a) Calcula el CPI del programa y la frecuencia de la CPU (usa el prefijo del	sistema internacional más adecuado).
CPI = 6x10^9 ciclos / 4,8x10^9 instrucciones = 1,25 c/i	
Frec = 6x10^9 ciclos / 2 segundos = 3x10^9 ciclos/segundo = 3 GHz	
b) <b>Calcula</b> los MIPS y MFLOPS a los que se ejecuta el programa.	
MIPS=4,8x10^9 instrucciones/ 2 segundos / 10^6 = 2400 MIPS	
MFLOPS= 2,4x10^8 operaciones / 2 segundos / 10^6 = 120 MFLOPS	
El tiempo de ejecución usado en el primer apartado se corresponde al tiempo comando "time" de linux hemos obtenido que el tiempo de CPU represe programa (wall time). El 80% restante es tiempo de entrada/salida (acceso acceso al disco duro del sistema tarda 8 milisegundos, mientras que si los o acceso tardaría 10 microsegundos.	enta solo el 20% del tiempo total de os al disco duro concretamente). Cada
c) Calcula la ganancia de la parte de entrada/salida si los datos del program un disco duro.	ma estuviesen en un SSD en lugar de er
Ganancia = 8x10-3 segundos /acceso / 10x10-6 segundos /acceso = 800	
d) <b>Calcula</b> la ganancia total en el programa que se obtendría con el cambio	o de tipo de disco.
Ganancia = $1/((1-fm)+fm/gm) = 1/((1-0.8)+0.8/800) = 4,975$	

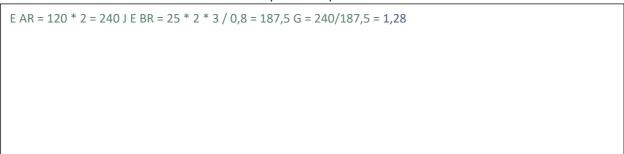
5 April 2018 12:11 pm 3 / 4

A pleno rendimiento, la CPU funciona a una frecuencia de 3 GHz y está alimentada a 1,6 V. En modo bajo consumo la CPU funciona a una frecuencia de 0,8 GHz y está alimentada a 1 V. Hemos medido que el consumo de la CPU en alto rendimiento es de 120W y en modo bajo consumo es de 25 W. En estos datos solo se considera la potencia debida a conmutación y la debida a fugas. Tanto la corriente de fugas (I) como la carga capacitiva equivalente (C) son las mismas en ambos modos.

e) **Calcula** la corriente de fugas (I) y la carga capacitiva equivalente (C) de la CPU (usar prefijo más adecuado del SI).



Calcula la ganancia en energía que tendría el sistema si ejecutara el programa en el modo de bajo rendimiento en vez de en el modo de alto rendimiento suponiendo que el CPI medio no varía.



5 April 2018 12:11 pm 4/4