

Cognoms: Nom:

1er Control Arquitectura de Computadors

Curs 2011-2012 Q2

Problema 1. (4 puntos)

Disponemos de una estación de trabajo con un procesador a 3.0 Ghz. Un programa de prueba P realiza 300 millones de operaciones de punto flotante. La ejecución de P tarda 1,5 segundos, y su CPI es 6 ciclos/instrucción.

a) **Calcula** el rendimiento en MIPS y MFLOPS de P.

b) **Calcula** el número de instrucciones ejecutadas por P.

Podemos considerar que en nuestro procesador sólo existen 2 tipos de instrucciones, las de coma flotante y las de enteros. Sabemos que las de enteros tienen un CPI de 3 c/i y que para realizar los 300 millones de operaciones de punto flotante se han ejecutado 500 millones de instrucciones de punto flotante.

c) **Calcula** el CPI de las instrucciones de coma flotante en P.

El fabricante del procesador está estudiando una modificación en el mismo que supondría una ganancia de 1,2 en las instrucciones de coma flotante y una ganancia de 0,75 en las instrucciones de enteros.

d) **¿Es beneficiosa esta mejora cuando ejecutamos P? Justificad la respuesta.**

Esta CPU, tiene una carga capacitiva equivalente de 12,5 nF (nanofaradios), y una corriente de fugas de 10 A y funciona a un voltaje de 1,2 V.

e) **Calcula** la potencia media debida a fugas, la debida a conmutación y la total para el programa P.

Este servidor está formado por los componentes mostrados en la tabla siguiente. La tabla también muestra el número de componentes de cada tipo y el tiempo medio hasta fallo (MTTF) de cada componente.

| Componente | Fuente alimentación | CPU | Ventilador CPU | Placa base | DIMMs | Disco duro |
|--------------|---------------------|-----------|----------------|------------|-----------|------------|
| Nº | 1 | 1 | 1 | 1 | 4 | 1 |
| MTTF (horas) | 100.000 | 1.000.000 | 100.000 | 200.000 | 1.000.000 | 100.000 |

El tiempo medio para reemplazar un componente que ha fallado (*mean time to repair*) es de 10 horas y la probabilidad de fallo sigue una distribución exponencial.

- f) **Calcula** el tiempo medio hasta fallos del hardware (MTTF), el tiempo medio entre fallos (MTBF) y la disponibilidad del sistema.

En los cálculos anteriores se ha supuesto que solo puede haber fallos de hardware, sin embargo hay otros aspectos que pueden influir en la fiabilidad de un sistema, como la estabilidad del software o la propia red eléctrica. Sabemos que el sistema operativo usado en nuestro servidor tienen un tiempo medio entre fallos de 15000 horas y la inestabilidad de la alimentación eléctrica (microcortes, caídas de tensión, etc) provoca un fallo cada 745 horas en media. Los fallos provocados tanto por el sistema operativo como la red eléctrica siguen también una distribución exponencial.

- g) **Calcula** el tiempo medio hasta fallos del sistema (MTTF) teniendo en cuenta la combinación del hardware, el SO y la red eléctrica. ¿cree que valdría la pena gastar mucho más en un computador cuyos componentes hardware tienen el doble de MTTF (respecto la tabla anterior)? Justifica la respuesta.

Apartado para nota: Cuando se produce un fallo (sea causado por el hardware, el SO o por un fallo eléctrico), se pierden los datos de la aplicación que se estaba ejecutando en ese momento y hay que volverla a ejecutar una vez el computador vuelve a estar en funcionamiento. En nuestro servidor queremos ejecutar una aplicación que tarda 1000 horas en ejecutarse, si durante su ejecución hay un fallo se deberá ejecutar de nuevo desde el principio. Por simplicidad supondremos que el tiempo de poner en marcha de nuevo el computador y la aplicación es negligible (**MTTR = 0**)

- h) **Calcula** la probabilidad de que el computador acabe la ejecución de la aplicación sin que se produzca ningún fallo. **Calcula** cuantas veces (en media) habrá que ejecutar la aplicación hasta que acabe completamente.

Cognoms: Nom:

1er Control Arquitectura de Computadors

Curs 2011-2012 Q1

Problema 2. (3 puntos)

Dada el siguiente código escrito en C:

```
typedef struct {                int Test(ss X1, ss *ps) {  
    int a;                      int i;  
    char ca;                    ss aux;  
    unsigned short us[5];      ...  
    char *p;                   }  
    char cb;  
} ss;
```

- a) **Dibujad** como quedaría almacenada la estructura de datos en linux, indicando claramente los desplazamientos respecto al inicio de la estructura y el tamaño de ésta. Dibujad el bloque de activación de la rutina Test.

En las siguientes preguntas, suponed que el código en C está dentro de la rutina Test.

- b) **Escribid** una secuencia de 3 instrucciones x86 que realice la siguiente asignación:

ps->a = i;

- c) **Escribid** una secuencia de 2 instrucciones x86 que realicen la siguiente asignación:

aux.p = &X1.cb;

d) **Escribid** una secuencia de 3 instrucciones x86 que realicen la siguiente asignación:

e) `ps->us[i] = ps->us[i] + 1;`

f) **Traducid** a x86 la siguiente asignación:

`i = Test(aux, &X1);`

Cognoms: Nom:

1er Control Arquitectura de Computadors

Curs 2011-2012 Q1

Problema 3. (3 puntos)

Dada el siguiente código escrito en C:

```
int Unknown(int v[], char *x);  
int Examen(int v[], char c, int *N){  
    int i;  
  
    if (c<'a' || c>'z')  
        i = *N;  
    else  
        i = 0;  
    return (i+Unknown(&v[0], &c));  
}
```

a) **Dibujad** el bloque de activación de la subrutina examen.

b) **Traducid** la subrutina examen a ensamblador del x86: