Jiabo Wang AC

PROBLEMA 12

Componente	Fuente de alimentación	CPU	Placa base	DIMM	GPU	Disco duro
Nº	1	1	1	4	1	8
MTTF (horas)	125000	1000000	200000	1000000	500000	100000

a) Calcula el tiempo medio hasta fallos del sistema

$$\frac{1}{MTTF_{sistema}} = \frac{1}{MTTF_{fuente}} + \frac{1}{MTTF_{cpu}} + \frac{1}{MTTF_{placa}} + \frac{1}{MTTF_{DIMM}}(4) + \frac{1}{MTTF_{GPU}} + \frac{1}{MTTF_{disco}}(8) =$$

$$\frac{1}{125000} + \frac{1}{1000000} + \frac{1}{200000} + \frac{4}{1000000} + \frac{1}{500000} + \frac{8}{100000} = \frac{1}{10000} = 10000 \text{ horas}$$

b) El tiempo medio para reemplazar un componente que ha fallado (MTTR) es de 20 horas. Calcula el tiempo medio entre fallos (MTBF).

$MTBF = MTTF + MTTR = 10000 + 20 = 10020$ horas

c) ¿Cual es la disponibilidad del sistema?

$$\text{Disponibilidad} = \frac{MTTF}{MTTF+MTTR} = \frac{10000}{10020} = 0.998$$

Problemas tema 2

PROBLEMA 1

Suponed que x e y, variables de tamaño 1 byte, tienen los valores 0x66 y 0xa3.

Expresión (bit a bit)	Valor binario	Valor hexadecimal
x & y	00000010	0x02
x | y	11110111	0xF7
~x | ~y	11111101	0xFD
x & !y (lógico)	00000000	0x00
x && y	00000001	0x01
x || y	00000001	0x01
!x || !y	00000000	0x00
x && ~y	00000001	0x01

PROBLEMA 2

x		x << 4		Para O (lógico) x >> 3		(aritmético) x >> 3	
hex	binario	hex	binario	hex	binario	hex	binario
0XF0	1111 0000	0x00	0000 0000	0x1E	0001 1110	0XFE	1111 1110
0x0F	0000 1111	0XF0	1111 0000	0x01	0000 0001	0x01	0000 0001
0xCC	1100 1100	0xC0	1100 0000	0x19	0001 1001	0XF9	1111 1001
0x55	0101 0101	0x50	0101 0000	0x0A	0000 1010	0x0A	0000 1010
0x80	1000 0000	0x00	0000 0000	0x10	0001 0000	0XF0	1111 0000
0x02	0000 0010	0x20	0010 0000	0x00	0000 0000	0x00	0000 0000

PROBLEMA 5

```
char A[256];
char tabla[256];

for (i=0; i<256; i++)
  A[i] = tabla[A[i]];
```

→

```
movl $A, %eax
movl $tabla, %ecx
movl $0, %ebx
for: cmpl $250, %ebx
     jge fi-for
     movsbl (%eax, %ebx), %edx   // %edx ← A[i]
     movb (%ecx, %edx), %dl      // %dl ← tabla[A[i]]
     movb %dl, (%eax, %ebx)      // A[i] ← tabla[A[i]]
     incl %ebx                   // ++i
     jmp for
fi-for:
```

PROBLEMA 6

```
int *sorpresa (int i, int *x) {
  if (i > -10 && i < 10)
    *x = i;
  else
    x = &i;
  return x;
}
```

i en 8(%ebp)

x en 12(%ebp)

→

```
sorpresa: push %ebp
          movl %esp, %ebp
          movl 8(%ebp), %ebx
          movl 12(%ebp), %ecx
          cmpl $-10, %ebx
          jle else
          cmpl $10, %ebx
          jge else
          movl %ebx, (%ecx)
          jmp fi
else:     leal 8(%ebp), %ebx
          movl %ebx, 12(%ebp)
fi:       movl 12(%ebp), %eax
          popl %ebp
          ret
```