

## AC Sesión 2: 1.12, 2.1, 2.2, 2.5, 2.6

1.12 a) Tiempo medio hasta fallos:  $MTTF_{sistema} = \frac{1}{\frac{1}{MTTF_1} + \frac{1}{MTTF_2} + \dots + \frac{1}{MTTF_N}}$

$$MTTF_{sistema} = \left( \frac{1}{125000} + \frac{1}{1000000} + \frac{1}{200000} + \frac{1}{1000000} + \frac{1}{500000} + \frac{1}{1000000} \right)^{-1} =$$

$$= 10.000 \text{ horas}$$

b) MTTR (Mean Time To Repair) = 20 horas  
MTBF?

$$MTBF = MTTF + MTTR = 10.000 + 20 = 10.020 \text{ horas}$$

c) disponibilidad (Availability) =  $\frac{MTTF}{MTTF + MTTR} = \frac{10.000}{10.000 + 20} = 0,992$  ? unidades?

2.1	expresión	binario	hex	expresión	binario	hex
	$x \& y$	00000010	0x02	$x \& \& y$	00000001	0x01
	$x   y$	11110111	0xF7	$x     y$	00000001	0x01
	$\sim x   \sim y$	11111101	0xFD	$! x   ! y$	00000000	0x00
	$x \& ! y$	00000000	0x00	$x \& \& \sim y$	00000000	0x00

$$\begin{cases} x = 0x66 = 01100110 \\ y = 0x93 = 10010011 \end{cases}$$

## AC Sesión 2

2.2

X		X << 4		(lógico) X >> 3		(aritmético) X >> 3	
hex	binario	hex	binario	hex	binario	hex	binario
0xF0	11110000	0x00	00000000	0xE	00011110	0xFE	11111110
0x0F	00001111	0xF0	11110000	0x01	00000001	0x01	00000001
0xCC	11001100	0xC0	11000000	0x11	00010001	0xF9	11111001
0x55	01010101	0x50	01010000	0x0A	00001010	0x0A	00001010
0x80	10000000	0x00	00000000	0x10	00010000	0xF0	11110000
0x02	00000010	0x20	00100000	0x00	00000000	0x00	00000000

2.5

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

```
movl $A,%eax //A
movl $tabla,%ebx //tabla
movl $0,%ecx //i=0
for: cmpl $256,%ecx //compara contador con
//immediato 256
jge fi_for
movl (%eax,%ecx),%edx //A[i] -> %edx
movb (%ebx,%edx),%al //tabla[A[i]] -> %al
movb %al, (%eax,%ecx) //A[i] = tabla[A[i]]
incl %ecx //++i
jmp for
```

fi\_for:

## AC Sesión 2

```

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

i = 8 (%ebp)
x = 12(%ebp)

```

??

sorpresal:

```

pushl %ebp           // %esp-4 → %ebp
                     // %ebp → M[%ebp]
movl %esp, %ebp
movl 8(%ebp), %eax    // i → %eax
movl 12(%ebp), %ebx   // x → %ebx
cmpl $-10, %eax
jle else              // i > -10
cmpl $10, %eax        // i < 10
jge fi_sorpresal
movl %eax, (%ebx)     // M[x] = i // *x = i
jmp fi_sorpresal
else:
leal (%ebx), %ebx     // M[x] = x → x
                     // x = &i
fi_sorpresal: movl 12(%ebp), %eax // x → %eax
popl %ebp             // %ebp ← M[%ebp]
                     // %esp ← %ebp+4
ret

```