PROBLEMAS S2:

1.12)

a) $$\frac{1}{MTTF_{sys}} = \frac{1}{MTTF_{cpu}} + \frac{1}{MTTF_{fan}} + \frac{1}{MTTF_{case}} + \frac{1}{MTTF_{gpu}} + \frac{4}{MTTF_{DIMM}} + \frac{8}{MTTF_{disco}} =$$

$$= \frac{1}{125 \cdot 10^3} + \frac{1}{10^6} + \frac{1}{200 \cdot 10^3} + \frac{4}{10^6} + \frac{1}{500 \cdot 10^3} + \frac{8}{100 \cdot 10^3} = \frac{1}{10000} \Rightarrow MTTF_{sys} = \frac{1}{1/10000} = \boxed{10\,000 h}$$

b) $MTBF = MTTF + MTTR = 10.000 + 20 = \boxed{10.020 h}$

c) $Availability = \dfrac{MTTF}{MTTF + MTTR} = \dfrac{10000}{10020} = \boxed{0.998}$

2.1) $x = 0x66$, $y = 0x93$

Expresión	Valor binario	valor hexa	Exp.	valor bin	valor hexa
x & y	0000 0110 1001 0011 $\boxed{000\ 0010}$	0x02	x && y	0000 0001	0x01
x | y	1111 0111	0xF7	x || y	0000 0001	0x01
~x | ~y	1111 1101	0xFD	!x || !y	0000 0000	0x00
x & !y	0000 0000	0x00	x && ~y	0000 0001	0x01

2.2)

x		x<<4		x>>3 (lógico)		x>>3 (aritmético)	
hex	bin	hex	bin	hex	bin	hex	bin
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

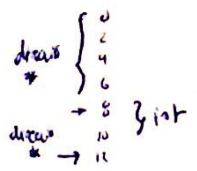
2.5) char A [256]
char tabla [256]

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

```
     movl $A, %eax  // %eax ← A
     movl $tabla, %ebx // %ebx ← tabla
     movl $0, %ecx  // %ecx ← i
for: cmpl $256, %ecx  // i<256
     jge fifor          // salto si i≥256
     movsbl {%eax,%ecx}, %edx  // %edx ← Extsign(A[i])
     movb {%ebx,%edx}, %dl  // %dl ← tabla(A[i]) 8 menor bits
     movb %dl, {%eax,%ecx}  // A[i]=tabla[i]"
     incl %ecx              // ++i
     jump for
fifor:
```

2.6)

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



```
sorpresa: pushl %ebp   // deixo espai * sorpresa
                          apunto al següent elem.
          movl %esp, %ebp  // %ebp ← esp
          movl 8(%ebp), %ebx  // %ebx ← 8(%ebp) [i]
          movl 12(%ebp), %ecx // %ecx ← 12(%ebp) (*x)

          cmpl $-10, %ebx      // i>-10
          jle else             // salto si i≤-10
          cmpl $10, %ebx       // i<10
          jge else             // salto si i≥10
          movl %ebx, (%ecx)    // x* = i
          jump return

else:     leal 8(%ebp), %ebx  // %ebx ← &i
          movl %ebx, 12(%ebp) // x = &i

return:   movl 12(%ebp), %eax // return x
          popl {%ebp}          // allibero pila.
          ret                  // eip ← M(%esp)
                                  %esp ← %esp+4
```

Escaneado con CamScanner