Apellidos: _____ **Nombre:**_____ **DNI:**_____ **Grupo:** _____

1º de Grado en Ingeniería Informática – COMPLETO

Final de enero de Fundamentos de Computadores (temas 1, 2, 3, 4, 5 y 6)

8 de enero 2017

Instrucciones para realizar el examen (tipo A)

- El tiempo disponible es de 3 horas.
- No olvide poner los apellidos y el nombre tanto en la hoja de examen como en los folios entregados.
- Para las preguntas tipo test, selecciónese una única respuesta en cada cuestión en el lugar habilitado para ello (señalando con una X en la tabla colocada al comienzo del test). El resto de preguntas se contestarán en folios.
- Cada dos respuestas incorrectas en el test anulan una correcta. Una pregunta sin contestar ni suma ni resta.
- Entregad tanto el enunciado del examen como los folios utilizados o no al acabar el examen.

Parte I: tipo test *(30%; 0.12 puntos por respuesta)*

A	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25
a																									
b																									
c																									

T1. La unidad de control:
a) Se encarga de decodificar las instrucciones y establecer las señales que dirigen la operación del camino de datos.
b) Es uno de los tres bloques en que se divide un computador según el esquema de Von Neumann, junto con la memoria y la entrada/salida.
c) Todas las respuestas anteriores son correctas.

T2. Indica cuál de los siguientes órdenes de capacidades de almacenamiento, de mayor a menor, es el único correcto:
a) 10 TB > 1 PB > 100 MB.
b) 1 TB > 5000 GB > 10000 MB.
c) 0,5 TB > 90000 MB > 10 GB.

T3. Una dirección hardware de interfaz de red (dirección MAC) tiene 48 bits. Eso significa que el máximo número de direcciones MAC distintas será de, aproximadamente:
a) Algo más de 256 billones (1 billón = 10^{12}) de direcciones.
b) Algo más de 4000 millones de direcciones.
c) Algo más de 2^{24} billones (1 billón = 10^{12}) de direcciones.

T4. El estándar de codificación de caracteres Latin1 es también conocido como:
a) UNICODE.
b) ISO 8859-1.
c) UTF16.

T5. En el estándar IEEE 754 de doble precisión, los bits de redondeo y retenedor ocuparán respectivamente:
a) Las posiciones -51 y -52 de la mantisa normalizada.
b) Las posiciones -53 y -54 de la mantisa normalizada.
c) Las posiciones -52 y -53 de la mantisa normalizada.

T6. Sea $F(A,B,C) = \sum m(1,2,7)$. Dicha función lógica equivale a:
a) $F(A,B,C) = A\cdot B\cdot \overline{C} + A\cdot \overline{B}\cdot C + \overline{A}\cdot \overline{B}\cdot \overline{C}$
b) $F(A,B,C) = (A+B+\overline{C})\cdot(\overline{A}+B+\overline{C})\cdot(A+B+C)$
c) Ninguna de las anteriores respuestas es correcta.

T7. Al realizar el OR lógico entre los bytes 0x7E y 0x81, obtenemos un número de 8 bits que expresado en C2 y traducido a decimal es:
a) 0
b) -1
c) 255

T8. En la representación de imágenes, el modelo de representación del color RGB:
a) Se utiliza principalmente para la impresión con tinta.
b) Es un modelo sustractivo, la suma de colores genera el negro.
c) Ninguna de las anteriores es correcta.

T9. En relación a las memorias ROM, señala la única opción verdadera:
a) En una EEPROM, el borrado se hace irradiando el circuito con una luz ultravioleta.
b) Si la ROM contiene 32 posiciones de 4 bits cada una, entonces tiene 8 líneas de entrada.
c) El plano AND se corresponde con un decodificador que selecciona la palabra de memoria a leer.

T10. Las funciones de un sistema operativo son:
a) Chequear que memoria, CPU y dispositivos funcionen bien al arrancar el ordenador.
b) Administrar CPU, memoria y dispositivos.
c) Administrar la correcta transferencia de datos entre el puente norte y puente sur, y entre los distintos dispositivos conectados a ellos.

T11. En cuanto a la potencia o características de la máquina o a su propósito principal de cómputo, los sistemas operativos se pueden clasificar en:
a) Linux, Windows o Mac OS.
b) De propósito general, de servidores, en tiempo real, integrados, para supercomputadoras...
c) De código abierto o de pago.

(sigue atrás)

T12. En Linux, si quisiera buscar los archivos ejecutables de los comandos más habituales, p.e., `ls`, `ps`, `rm`, `cp`, etc., los encontraría en el directorio:
a) /bin/.
b) /usr/bin/.
c) /etc/.

T13. La diferencia entre un enlace físico (*duro*) y uno simbólico (*blando*) radica en que:
a) El enlace duro "gastará" tanto espacio de más en disco duro cuanto sea el tamaño de los datos del fichero original enlazado (no así el enlace blando).
b) El enlace simbólico no permite modificar el contenido del fichero sino solamente modificando el fichero enlazado original (no así el enlace duro).
c) Podemos borrar cualquier enlace duro que, mientras que subsista uno, los datos serán accesibles, mientras que, en los enlaces blandos, si borramos el fichero enlazado original, los datos no serán accesibles.

T14. Una expresión que englobara a los nombres de un fichero que empezara por a, con tantos caracteres como fueren con la única condición que acabara en z antes de un punto, al que le siguieran dos o más caracteres, sería:
a) `a?z.**`
b) `a*z.??`
c) `a*z.?*?`

T15. Si, en Linux, queremos que se nos muestren todos los archivos (no directorios) del directorio `/tmp` y sus descendientes cuyo nombre sea exactamente "`examen.pdf`", tendremos que ejecutar la orden:
a) `ls -lR /tmp -type f -name "examen.pdf"`
b) `find /tmp -type f -name "examen.pdf"`
c) `find -r /tmp -type f -iname "examen.pdf"`

T16. ¿Cuál de estas frases acerca de las librerías estáticas y dinámicas en Linux es correcta?
a) Un binario (ejecutable) que ha sido enlazado dinámicamente responde a la orden `ldd` informando de las librerías enlazadas.
b) Un binario (ejecutable) que ha sido enlazado estáticamente responde a la orden `ldd` informando de las librerías enlazadas.
c) Si no se indica lo contrario, el compilado y enlazado por defecto de un archivo fuente en lenguaje C, con la orden `gcc`, es de forma estática.

T17. La orden Linux que muestra todos los procesos en ejecución actualizándose en tiempo real es:
a) `top`
b) `ps -Af`
c) `fg`

T18. La instrucción `movq $0,(%rax)` del ensamblador del x86-64 sirve para:
a) Almacenar el valor cero en las 8 direcciones consecutivas de memoria apuntadas por la dirección contenida en el registro `%rax`.
b) Almacenar el valor cero en los 4 bytes consecutivos de la dirección de memoria contenida en el registro `%rax`.
c) Copiar el contenido del registro `%rax` de memoria al registro `$0` del procesador.

T19. Para apilar el contenido del registro `%eax` usaríamos el código ensamblador del x86-64 siguiente:
a) pushl %eax.
b) movl %eax, (%esp) y después subl $4, %esp.
c) Las dos formas serían válidas.

T20. Un compilador:
a) Traduce un código fuente escrito en un lenguaje de alto nivel a código ejecutable final.
b) Compila un módulo objeto con librerías estáticas o dinámicas para conseguir el ejecutable final.
c) Es un módulo del sistema operativo que es llamado por el cargador y que ayuda a éste a preparar la pila.

T21. La instrucción `ret` del repertorio del x86-64:
a) Es funcionalmente idéntica a una instrucción `jmp`.
b) Salta a la dirección contenida en la cima de la pila.
c) Salta a la dirección contenida en el registro `%rip`.

T22. En lo referente al ensamblador del x86-64:
a) Una etiqueta únicamente puede ser sinónimo de una dirección de memoria del segmento de instrucciones.
b) Una etiqueta únicamente puede ser sinónimo de una dirección de memoria del segmento de datos.
c) Una etiqueta es sinónimo de una dirección de memoria, bien del segmento de instrucciones o del de datos.

T23. Las siglas DHCP se refieren a:
a) La conversión de direcciones para permitir el uso de IPs privadas en subredes manteniendo la conectividad con Internet.
b) La determinación automática de los parámetros de la red en un host mediante un protocolo de configuración.
c) El servicio de traducción a/de nombres de host de/a direcciones IP.

T24. En redes, el termino NAT:
a) Significa Network Access Transport, y pertenece a la capa de transporte de la pila de protocolos.
b) Significa Network Address Translation y permite conectar varias interfaces usando una única dirección IP pública.
c) Significa National Associated Telematics y es un organismo que vela a nivel local por el buen uso de internet.

T25. Considerando varios procesos de red tipo cliente en una misma máquina, que se han conectado a un proceso único en un servidor. Es cierto que:
a) Cada proceso cliente emplea un puerto distinto para cada comunicación.
b) El cliente ha de tener varias direcciones IP para que cada proceso envíe sus mensajes de origen con una IP distinta.
c) Es el servidor el que, al estar replicado, ofrece varias direcciones IP para que el cliente utilice direcciones IP distintas para cada proceso.

Apellidos: _____ **Nombre:**_____ **DNI:**_____ **Grupo:** _____

1º de Grado en Ingeniería Informática – COMPLETO

Parte II: cuestiones teórico-prácticas *(32.5%; puntuación indicada en cada apartado)*

C1. (*1,0 puntos*) Considérese el número real **N=+5,5** (dado en decimal):

a) ¿Cuál es su representación en coma fija en base 2? (utilizar 8 bits en total para representar el número, 4 para la parte entera y 4 para la parte decimal)

b) ¿Cuál es su representación en un formato coma flotante normalizado IEEE 754 corto que usa sólo 8 bits en total, con 1 bit para el signo, 4 bits para el exponente y 3 bits para la mantisa? **Nota:** Emplear en este apartado un sesgo S=7, y dar la respuesta en hexadecimal.

Considérese ahora el siguiente byte de información: **01001110**:

c) Si representa un número natural (entero SIN signo), ¿cómo codificaría el ordenador la misma información en 2 bytes de tamaño? (dar la respuesta en hexadecimal).

d) Si representa un número entero CON signo y formato en complemento a 2, ¿cómo codificaría el ordenador la misma información en 2 bytes de tamaño? (dar la respuesta en hexadecimal).

e) Si representa un número entero CON signo y notación sesgada, ¿cómo codificaría el ordenador la misma información con 2 bytes de tamaño? (dar la respuesta en hexadecimal) **Nota.-** Emplear en este apartado un sesgo S=128 para 1 byte, y S=32768 para 2 bytes.

f) Si representa un carácter en ISO 8859-1, ¿de qué carácter se trata sabiendo que la letra 'A' (mayúscula) empieza en la posición 0x41 = $65)_{10}$?

g) Si el mismo carácter anterior lo queremos representar en UNICODE UTF-8, ¿cómo se representaría?

C2. *(0,75 puntos)* Indicar los nombres de las cinco capas de la arquitectura de red usada por Internet, ordenadas de más alto a más bajo nivel. A continuación, indicar dos protocolos diferentes pertenecientes cada una de las capas primera, segunda y cuarta, más un sólo protocolo para la tercera capa (entendiendo de nuevo las capas ordenadas de más alto a más bajo nivel). Sólo será necesario expresar el significado concreto de las siglas de los dos protocolos mencionados en la segunda capa.

C3. (*1,5 puntos*) Impleméntese un circuito que admita (mediante 4 entradas binarias) números del 1 al 12 que representan los meses del año (el cero y los valores del 13 al 15 serán considerados indeterminaciones en las salidas). La salida S_{31} del circuito ha de ponerse a 1 si sus entradas representan un mes de 31 días. La salida S_{30} ha de ponerse a 1 si sus entradas representan un mes de 30 días de duración. Para ello:

a) Realícese la tabla de verdad de ambas funciones.

b) Indíquese S_{31} minimizado según su expresión lógica como suma de productos.

c) Indíquese S_{30} minimizado según su expresión lógica como producto de sumas.

d) Realícese el circuito mínimo de S_{31} sólo con puertas NAND y el circuito mínimo de la salida S_{30} sólo con puertas NOR, y dibújense los dos circuitos resultantes.

e) Para la implementación del circuito S_{30} hay varias minimizaciones alternativas posibles. Dibújese una alternativa a la empleada en la solución al apartado c).

Parte III: ejercicios boletines *(37.5%; puntuación indicada en cada apartado)*

P1. *(1,0 puntos)* La siguiente imagen corresponde a un pantallazo de un archivo abierto con *okteta*:

```
0000:0000 496E6765 6E696572 ED612049 6E666F72 Ingeniería Infor
0000:0010 6DE17469 636122FF 00000000 0000F07F mática"ÿ......ð.
0000:0020 0000B2C1                            ..²Á
```

Se sabe que en él se archivaron, siguiendo el convenio *little endian*, los siguientes datos (en el orden indicado, y siempre un dato a continuación del otro):

1. Una cadena de caracteres.
2. Un número entero en complemento a 2 de 16 bits.
3. Un número real en IEEE 754 de doble precisión.
4. Un número real en IEEE 754 de simple precisión.

Se pide:

a) Indicar los desplazamientos de bytes (desde-hasta) ocupados por la cadena de caracteres, así como los caracteres primero y último, respectivamente, de dicha cadena. **Nota:** dicho tamaño se puede deducir del tamaño total ocupado por todo el archivo, igual a la suma de los tamaños de los cuatro datos indicados que lo forman en su totalidad.

b) ¿Qué codificación concreta crees que se ha utilizado para dicha cadena, ASCII, Latin1, UTF8 ó UTF16? Justifica brevemente tu respuesta.

c) Indicar los desplazamientos de byte (desde-hasta) ocupados por el número real en doble precisión, y obtener el valor concreto representado. **Pista:** nótese que se trata de un caso especial del estándar IEEE 754.

d) Indicar los desplazamientos de byte (desde-hasta) ocupados por el número real en simple precisión, y obtener el valor concreto representado (expresado en base 10).

P2. *(0,9 puntos)* Suponiendo un *prompt* del sistema `alumno@lab:~$`, qué ordenes Linux habría que ejecutar en un sólo paso para poder realizar las siguientes acciones?

Nota: Siempre que sea necesario utilizar una ruta ésta deberá ser RELATIVA.

a) Mostrar por el terminal el contenido de un fichero oculto llamado `.preguntas.txt` ubicado en `/home/alumno/ejercicios`.

b) Renombrar como `ejemplo2.txt` un fichero que se llama `ejemplo.txt` y está ubicado en `/home/alumno/fc/ejercicios/ejemplo.txt`.

c) Realizar un listado largo (con información de permisos, tamaños, etc.) del contenido de una carpeta `practicas` que cuelga de `/home/alumno/fc`.

d) Mover todo un directorio llamado `ANTIGUO` dentro de un directorio existente llamado `NUEVO`, ubicados ambos en `/home/alumno`.

e) Trasladar un archivo llamado `documento.pdf` ubicado en el directorio `/home/alumno/fc/practicas` al directorio presente.

f) Duplicar un directorio `Practicas`, con todas sus subcarpetas y archivos, que cuelga de `/home/alumno` en `/home/alumno/fc` , con el nuevo nombre `Practicas_FC`.

g) Borrar totalmente el directorio `Practicas` del punto anterior, con todo su contenido.

h) Dado un achivo `foto1.jpg` dentro de una carpeta `/home/alumno/fotos`, dar únicamente permiso de lectura de ese archivo a `alumno` y a su grupo, y ningún permiso a nadie más.

i) Dar permisos a la carpeta anterior (`/home/alumno/fotos`) para que sólo `alumno` pueda crear y borrar archivos de ella, su grupo pueda acceder a los archivos del mismo, pero sin crear ni borrar archivos en ella, y el resto de usuarios no pueda acceder ni en lectura ni en escritura a dicha carpeta.

P3. *(1,0 puntos)* Considérese la siguiente sesión con el depurador `gdb` (aparecen subrayados los comandos tecleados por el usuario; el resto es la salida producida por el terminal), y rellenar todos los huecos del texto que va a continuación (indicando en el examen la correspondiente referencia al hueco para cada respuesta):

```
user@host:~/$ gdb ./main
(gdb) list
1   int array[4] = {-2,-6,-8,-14};
2   int i;
3   int main() {
4         for(i=0;i<4;i++)
5         array[i] = funcion(i);
6   }
7   int funcion(int parametro) {
8      return array[parametro] / 2;
9   }
(gdb) disassemble funcion
   0x0000000000400625 <+0>:        push   %rbp
   0x0000000000400626 <+1>:        mov    %rsp,%rbp
   0x0000000000400629 <+4>:        mov    %edi,-0x4(%rbp)
   0x000000000040062c <+7>:        mov    -0x4(%rbp),%eax
   0x000000000040062f <+10>:       cltq
   0x0000000000400631 <+12>:       mov    0x601070(,%rax,4),%eax
   0x0000000000400638 <+19>:       mov    %eax,%edx
   0x000000000040063a <+21>:       shr    $0x1f,%edx
   0x000000000040063d <+24>:       add    %edx,%eax
   0x000000000040063f <+26>:       sar    %eax
   0x0000000000400641 <+28>:       pop    %rbp
   0x0000000000400642 <+29>:       retq
(gdb) x/29bx 0x400625
0x400625 <funcion>:    0x55  0x48  0x89  0xe5  0x89  0x7d  0xfc  0x8b
0x40062d <funcion+8>:  0x45  0xfc  0x48  0x98  0x8b  0x04  0x85  0x70
0x400635 <funcion+16>: 0x10  0x60  0x00  0x89  0xc2  0xc1  0xea  0x1f
0x40063d <funcion+24>: 0x01  0xd0  0xd1  0xf8  0x5d
```

```
(gdb) x/16xb array
0x601070 <array>:      0xfe  0xff  0xff  0xff  0xfa  0xff  0xff  0xff
0x601078 <array+8>:    0xf8  0xff  0xff  0xff  0xf2  0xff  0xff  0xff
```

"El código depurado en la sesión anterior manipula una tabla de **__($P_3$1)__** *elementos de tipo entero, que ocupará exactamente* **__($P_3$2)__** *bytes en memoria, y que comienza en la dirección exacta* **__($P_3$3)__** *. La función `funcion`, una vez ubicada en memoria, comienza exactamente en la dirección* **__($P_3$4)__** *. Una vez ejecutado el código, el nuevo valor del byte que ocupa la dirección 0x601074 será* **__($P_3$5)__** *, en lugar de su valor inicial, que era* **__($P_3$6)__** *(expresar ambos bytes en formato 0xXX, con XX en hexadecimal). Por otro lado, la instrucción que se ejecutará justo antes del `push rbp` (que no aparece en el listado mostrado, pero que se puede fácilmente deducir) será* **__($P_3$7)__** *, mientras que la instrucción tras cuya ejecución se volverá después para seguir ejecutando el programa por la instrucción siguiente a la anteriormente indicada será* **__($P_3$8)__** *. Las instrucciones utilizadas para guardar en la pila el valor anterior del registro base de la pila, antes de modificarlo en la función, y posteriormente recuperarlo serán, respectivamente,* **__($P_3$9)__** *y* **__($P_3$10)__**. *El registro que en todo momento contiene la dirección de la cima de la pila es el registro* **__($P_3$11)__** *. Finalmente, el código máquina de la instrucción `cltq` ocupará exactamente* **__($P_3$12)__** *bytes en memoria, cuyos valores exactos son* **__($P_3$13)__** *, y se sitúan en el rango de direcciones de byte comprendido entre la dirección* **__($P_3$14)__** *y la* **__($P_3$15)__** *, ambas inclusive."*

P4. *(0,85 puntos)* Rellene los huecos de cada apartado sobre redes de ordenadores:

a) Dado un equipo con IP y máscara `155.54.202.190/22`, su dirección de red es **__(P_4a1)__** y su máscara de red es **__(P_4a2)__**.

b) En dicho equipo se han ejecutado dos comandos Linux. Rellene los huecos considerando que su router por defecto tiene asignada la primera IP (es decir, la dirección válida más baja) disponible de la subred:

```
user@host:~$ __(P4b1)__ eth0
eth0  Link encap:Ethernet  direcciónHW 00:00:10:11:12:13
Direc. inet:155.54.202.190  Difus.: __(P4b2)__ [...]

user@host:~$ __(P4b3)__
Destino   Pasarela      Genmask      Indic  Métric  Ref  Uso  Interfaz
_(P4b4)_  0.0.0.0    255.255.252.0   U      100     0    0    eth0
default   _(P4b5)_      0.0.0.0      UG     100     0    0    eth0
```

c) Rellene los huecos de cada apartado (c1), (c2) y (c3): *"Si dispongo de una red global `155.54.1.0/24` tendré un total (incluyendo posibilidad de router) de hasta* **_(P_4c1)_** *interfaces de red; si configurara entonces todos los equipos de la misma con la máscara `255.255.255.224`, obtendría* **_(P_4c2)_** *subredes de* **_(P_4c3)_** *interfaces de red (incluyendo los respectivos routers) cada una."*

Apellidos: _____ **Nombre:**_____ **DNI:**_____ **Grupo:** _____

Soluciones a EXAMEN COMPLETO:

Test:

EXAMEN COMPLETO A

A	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20	T21	T22	T23	T24	T25
a	X		X									X				X	X	X	X	X					X
b				X	X		X			X	X				X						X		X	X	
c		X				X		X	X				X	X								X			

Cuestiones:

C1: *(1,0 puntos)*

a) $+5{,}5)_{10}$ = **0101,1000**$)_2$
b) $0101{,}1000)_2 = 1{,}011)_2 \cdot 2^2$; M = 011; exp = 2 + 7 = 9 = $1001)_2$ => $n^o{}_{IEEE\text{-}754}$ = 0 1001 011 = 0100 1011 = 0x**4B.**
c) 0000 0000 0100 1011 = **0x004E**
d) 0000 0000 0100 1011 = **0x004E**
e) 0x4E = $78)_{10}$ → 78 – 128 = -50 → -50 + 32768 = 32718 = 0x7FCE
f) $4E)_{16} – 41)_{16} = 13)_{10}$; 'A' + 13 = 'N'
g) 4E = 0100 1110; En codificación UTF-8 coincide con el código ASCII de 7 bit → 4E.

C2: *(0,75 puntos)*

Transparencias de teoría tema 6, página 16:
1. Capa de Aplicación, protocolos HTTP y SMTP (por ejemplo).
2. Capa de Transporte, protocolos TCP (Transmission Control Protocol) y UDP (User Datagram Protocol).
3. Capa de Red, protocolo IP.
4. Capa de Enlace, protocolos IEEE 802.3 (Ethernet) y 802.11 (WiFi) (por ejemplo).
5. Capa Física.

C3: *(1,5 puntos)*

a)

minterm	E_3	E_2	E_1	E_0	S_{31}	S_{30}
0	0	0	0	0	-	-
1	0	0	0	1	1	0
2	0	0	1	0	0	0
3	0	0	1	1	1	0
4	0	1	0	0	0	1
5	0	1	0	1	1	0
6	0	1	1	0	0	1
7	0	1	1	1	1	0
8	1	0	0	0	1	0
9	1	0	0	1	0	1
10	1	0	1	0	1	0
11	1	0	1	1	0	1
12	1	1	0	0	1	0
13	1	1	0	1	-	-
14	1	1	1	0	-	-
15	1	1	1	1	-	-

S_{31}

$E_1 E_0$ \ $E_3 E_2$	00	01	11	10
00	-	0	1	1
01	1	1	-	0
11	1	1	-	0
10	0	0	-	1

S_{30}

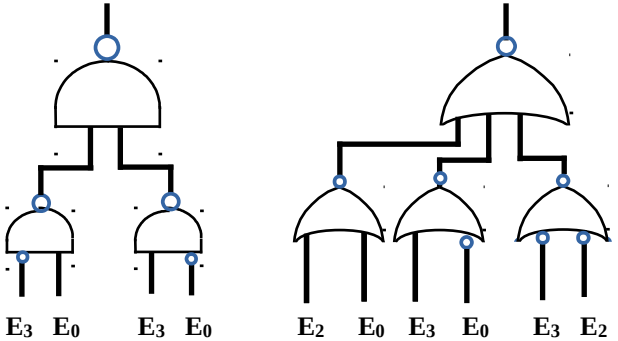
$E_1 E_0$ \ $E_3 E_2$	00	01	11	10
00	-	1	0	0
01	0	0	-	1
11	0	0	-	1
10	0	1	-	0

b) $S_{31} = (\overline{E}_3 \cdot E_0) + (E_3 \cdot \overline{E}_0)$

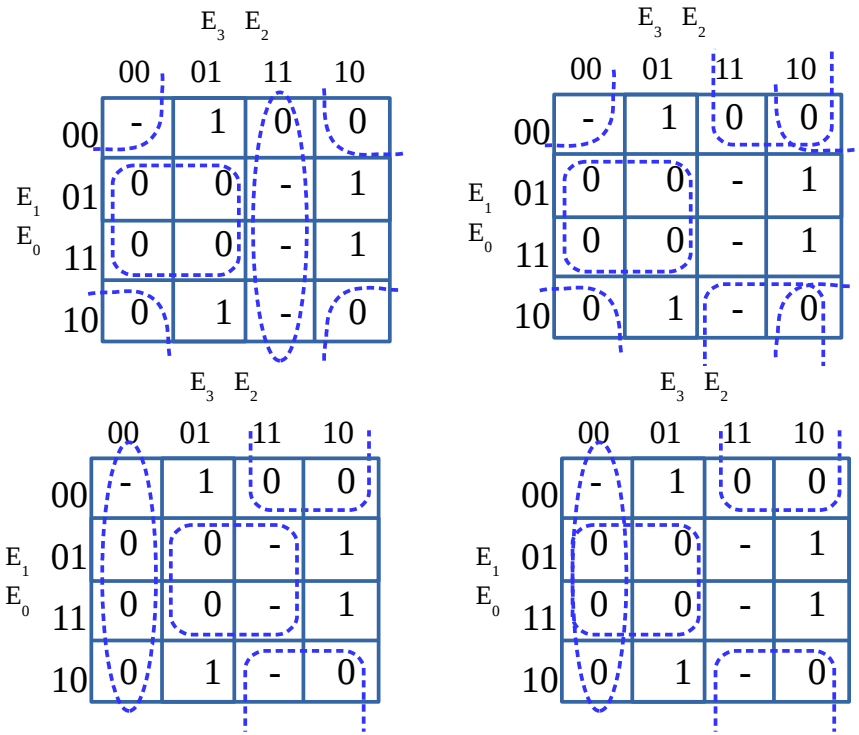
c) $S_{30} = (E_2 + E_0) \cdot (E_3 + \overline{E}_0) \cdot (\overline{E}_3 + \overline{E}_2)$ (Por ejemplo)

Nota: Para S_{31} todos los grupos resultantes son implicantes primos esenciales, con lo que la implementación mínima es única. Para S_{30} existen varias coberturas igualmente mínimas, de las que se ha cogido una cualquiera.

d) Aplicando Morgan en uno y otro caso, las expresiones c) y d) quedan (expresadas ya como circuitos):



e) Posibles soluciones:



Prácticas:

P1: *(1,0 puntos)*

a) Desplazamientos **0000:0000 a 0000:0015**, ambos inclusive. El primer carácter es la **'I'** (byte 0x49), y el último la **'a'** (byte 0x61).

b) Se está usando la codificación de caracteres **Latin1**, ya que es la única de las cuatro mencionadas en las que los caracteres acentuados 'í' y 'á' ocupan exactamente 1 byte. El resto de caracteres no acentuados presentes en la cadena también ocupan 1 sólo byte. Dado que en ASCII puro no existen dichos caracteres acentuados; que en UTF16 bits todos los caracteres, no sólo los acentuados, ocuparán 16 bits, es decir, dos bytes; y que en UTF8, aunque los caracteres no acentuados ocupan 1 byte, los acentuados ocuparán siempre dos bytes, por eliminación, nos queda que la codificación utilizada para la cadena ha de ser forzosamente Latin1, también llamada ISO-8859-1 (o ISO-8859-15, si añadiese el símbolo del euro, cosa que no sabemos a priori a la vista del pantallazo, pero que tampoco hace falta para resolver el ejercicio).

c) El entero de 16 bits en C2 ocupará los desplazamientos 0000:0016 a 0000:0017, que nos saltamos, para llegar a los desplazamientos **0000:0018 a 0000:001F**, ambos incluidos. Se trata de los bytes 0x7FF0000000000000, una vez agrupados ya en 64 bits siguiendo el convenio *little endian*. Esto se corresponde, en bits, a la ristra:
0x7FF0000000000000 → 0111 1111 1111 00.... → 0 11111111111 00000.... →
signo positivo, campo exponente todo unos, y campo mantisa todo ceros → se trata del caso especial +∞ (**más infinito**).

d) Finalmente, el número real en IEEE 754 de simple precisión se encuentra en los desplazamientos **0000:0020 a 0000:0023** ambos inclusive. Se trata de los bytes 0x00 0x00 0xB2 0xC1. Por estar en *little endian*, quedan en 32 bits como 0xC1B20000. Al interpretarlos en IEEE 754 de simple precisión:

0xC1B20000 → 1100 0001 1011 0010 0000 0000 0000 0000 →
1 10000011 01100100000000000000000
M = 1,m → M=1,01100100000000000000000)$_2$ = 1,011001)$_2$
e = E + Sesgo = 10000011)$_2$ = 131 → E = 131 - 127 → E = 4
S = 1 (número negativo)
N = -1,011001)$_2$ * 2^4 = -10110,01$_{)2}$ = **-22.25**

Apellidos: _____ **Nombre:**_____ **DNI:**_____ **Grupo:** _____

P2: *(0,9 puntos)*

Los comandos siguientes constituyen una posible solución a cada apartado, aunque en algunos casos puede haber alguna otra solución igualmente válida:

a) `cat ejercicios/.preguntas.txt`
b) `mv fc/ejercicios/ejemplo.txt fc/ejercicios/ejemplo2.txt`
c) `ls -l fc/practicas`
d) `mv ANTIGUO NUEVO`
e) `mv fc/practicas/documento.pdf .`
f) `cp -r Practicas fc/Practicas_FC`
g) `rm -r Practicas`
h) `chmod 440 fotos/foto1.jpg`
i) `chmod 750 fotos/`

P3: *(1,0 puntos)*

$P_3$1) 4
$P_3$2) 16
$P_3$3) 0x601070
$P_3$4) 0x400625
$P_3$5) 0xfd
$P_3$6) 0xfa
$P_3$7) call funcion
$P_3$8) retq
$P_3$9) push %rbp
$P_3$10) pop %rbp
$P_3$11) %rsp
$P_3$12) 2
$P_3$13) 0x48 y 0x98
$P_3$14) 0x40062f
$P_3$15) 0x400630

P4: *(0,85 puntos)*

P_4a1) 155.54.200.0
P_4a2) 255.255.252.0

P_4b1) ifconfig
P_4b2) 154.54.203.255
P_4b3) route -n
P_4b4) 155.54.200.0
P_4b5) 155.54.200.1

P_4c1) 254
P_4c2) 8
P_4c3) 30