

Apellidos: _____ Nombres: _____

Ejercicio 1

El siguiente código de máquina y su desensamblado RISC-V **computa la suma prefijo en el mismo arreglo** (*in-place prefix sum*). El arreglo *a* está en el segmento ELF *.bss* y empieza en *0x2FC0* y termina en *0x3008* **exclusive**. Como sus elementos son *unsigned long*, cada uno ocupa 8 bytes y por lo tanto tiene 9 elementos.

00000000000000634 <main>:

```

634: 0613          li    a2,0x3008    # <__BSS_END__> &a[9]
636: b206          li    a5,0x2FC8    # <a+0x8> &a[1]
638: 6398          ld    a4,0(a5)    # a4 = a[i]
63a: ff87b683      ld    a3,-8(a5)    # a3 = a[i-1]
63e: 9736          add   a4,a4,a3
640: e398          sd    a4,0(a5)    # a[i] = a4
642: 07a1          addi  a5,a5,8      # "i++"
644: fec79ae3      bne   a5,a2,0x638    # <main+0x10>, "i<9"
648: 8082          ret
    
```

Escribir la **traza de memoria** completa que genera la ejecución del proceso **incluyendo los *instruction fetch***.

Inicialización

_____, _____,

Vuelta 1

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 2

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 3

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 4

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 5

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 6

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 7

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 8

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Fin

_____.

Ejercicio 2

Supongamos que en trampoline.S, la rutina en ensamblador RISC-V que guarda los registros de espacio de usuario se comete un pequeño error por culpa del gato 🐱. La parte que los restituye está perfecta.

sd a1, 120(a0)

sd a2, 128(a0)

sd a3, 136(a0)

sd a4, 144(a0)

sd a2, 152(a0) #ERROR!

sd a6, 160(a0)

sd a7, 168(a0)

ld a1, 120(a0)

ld a2, 128(a0)

ld a3, 136(a0)

ld a4, 144(a0)

ld a5, 152(a0)

ld a6, 160(a0)

ld a7, 168(a0)

Indicar en el código de máquina del Ejercicio 1, **ENTRE** qué líneas se puede producir un TRAP sin cambiar el funcionamiento del programa (poner "**TRAP**") y entre qué líneas ese TRAP resulta fatal para la ejecución de nuestro código (poner "**F**"). Recalcamos, poner TRAP ó F entre cada una de las líneas de código indicando si se puede producir un TRAP de manera inocua o no.

Ejercicio 3

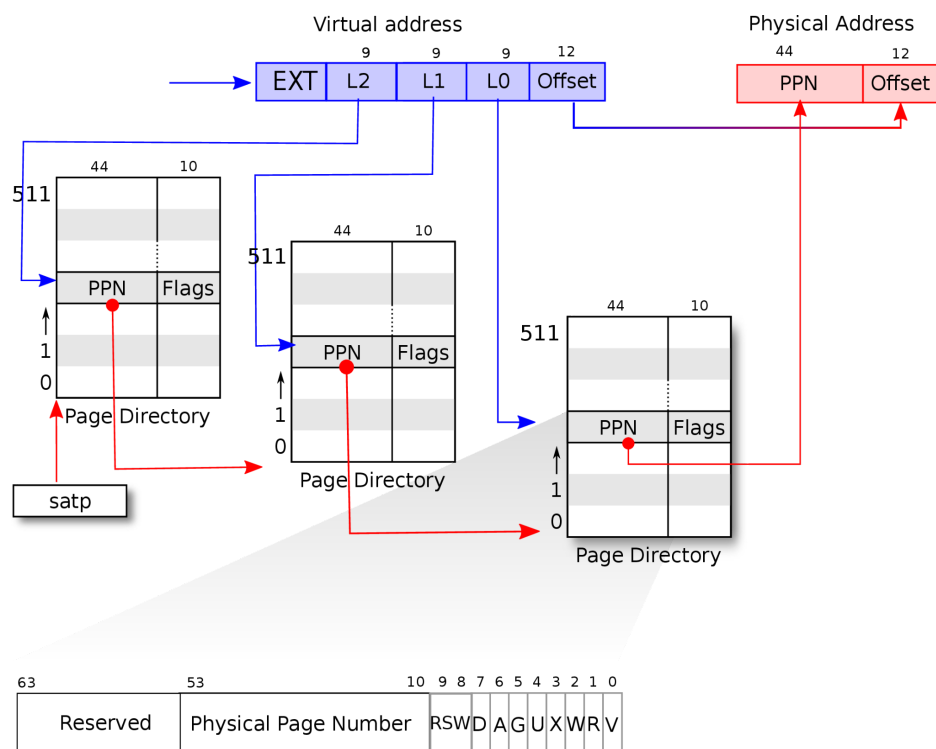
Planificar con Round Robin Q=2 para los siguientes procesos que tienen mezcla entre cómputo CPU y espera IO. Ante situaciones de simultaneidad, ordenar alfabéticamente, por ejemplo ¿Cuál de los tres procesos inicia en tiempo 0?: el "A".

Proceso	Inicio	CPU	IO	CPU
A	0	1	4	3
B	0	1	1	1
C	0	8		



Ejercicio 4

Tenemos un esquema de paginación RISC-V con páginas de 4 KiB de 3 niveles con formato 9,9,9,12 -> 44,12 como muestra la figura.



Bits de control

V: válido

R: se puede leer, *readable*

W: se puede escribir, *writable*

X: se puede ejecutar, *executable*

Supongamos que tenemos el registro de paginación apuntando al marco físico `satp=0x0000000FE0`.

<p>0x0000000FE0</p> <p>-----</p> <p>0x1FF: 0x0000000000, ----</p> <p>:</p> <p>0x004: 0x0000000000, ----</p> <p>0x003: 0x0000000000, ----</p> <p>0x002: 0x0000000FEA, XWRV</p> <p>0x001: 0x0000000FEA, XWRV</p> <p>0x000: 0x0000000FEA, XWRV</p>	<p>0x0000000FEA</p> <p>-----</p> <p>0x1FF: 0x0000000000, ----</p> <p>:</p> <p>0x004: 0x0000000000, ----</p> <p>0x003: 0x0000000000, ----</p> <p>0x002: 0x000000AD0BE, XWRV</p> <p>0x001: 0x000000AD0BE, XWRV</p> <p>0x000: 0x000000AD0BE, XWRV</p>	<p>0x000000AD0BE</p> <p>-----</p> <p>0x1FF: 0x0000000000, ----</p> <p>:</p> <p>0x004: 0x0000000000, ----</p> <p>0x003: 0x000000D1AB10, XWR-</p> <p>0x002: 0x000000DECADA, -WRV</p> <p>0x001: 0x000000CAFECAFE, ----</p> <p>0x000: 0x000000ABAD, X--V</p>
---	--	--

a) Traducir de **virtual a física** las direcciones:

0x0000

0x1000

0x2000

0x3000

b) Traducir de la dirección física 0xDECADEA980 a **todas** las virtuales que la apuntan.

Ejercicio 5

A la luz del esquema de paginación del Ejercicio 4, indicar de manera esquemática que es lo que pasaría con la traza de memoria respecto a la ejecución de un proceso corriendo el código de máquina del Ejercicio 1.

Inicialización

_____, _____,

Vuelta 1

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 2

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 3

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 4

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 5

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 6

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 7

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Vuelta 8

_____, _____, _____, _____, _____, _____, _____, _____, _____,

Fin

_____.