

Para todos los problemas del enunciado, si es necesario hacer suposiciones, deberán estar indicadas claramente, por escrito. Las mismas serán consideradas parte integral de tu respuesta en el momento de la corrección.

1. Pasar a assembly MIPS el siguiente fragmento de código, respetando la ABI presentada en clase. Deberá incluir en la resolución un diagrama del stack frame resultante.

```
size_t
strnlen(const char *s, size_t maxlen)
{
    size_t len;

    for (len = 0; len < maxlen; len++, s++) {
        if (!*s)
            break;
    }
    return (len);
}
```

2. Se tiene un procesador basado en MIPS32, que incluye una MMU (Memory Management Unit) con páginas de tamaño 2^{12} bytes, direcciones virtuales de 32 bits y direcciones físicas de 32 bits. Esta MMU utiliza política de reemplazo LRU al paginar (cuando debe intercambiar páginas físicas en RAM contra el disco). Se corre un programa de prueba y el mismo se detiene justo antes de la ejecución de las siguientes instrucciones en la dirección 0x1ffc:

```
lw $t0, 0x34c8($zero)      | PC = 0x1ffc
sw $t0, 0x6004($zero)      | PC = 0x2000
```

En el siguiente cuadro se observan las primeras 8 PTEs de la tabla de páginas al momento en que la ejecución es detenida, indicando las dos entradas menos recientemente utilizadas (LRU y next LRU). Asumir que todas las páginas en memoria física están en uso. Se reanuda la ejecución y se ejecutan las instrucciones `lw` y `sw`. Los bits de control D y P son Dirty y Present, respectivamente.

VPN	D	P	PPN
0	1	1	0x1
1	0	1	0x0
(LRU) 2	1	1	0x6
3	-	0	-
(next LRU) 4	0	1	0x4
5	0	1	0x2
6	0	1	0x7
7	0	1	0x3

- a) Mostrar el contenido de la tabla de páginas luego de completar la ejecución de `sw`.
- b) ¿Qué páginas físicas, si hay alguna, deben ser salvadas a disco durante la ejecución de estas instrucciones?
- c) Indicar las direcciones físicas asociadas a las cuatro direcciones virtuales accedidas durante la ejecución de estas instrucciones.
 - 1) dirección física de la instrucción `lw`: 0x_____
 - 2) dirección física del dato leído por la instrucción `lw`: 0x_____

- 3) dirección física de la instrucción `sw`: `0x_____`
4) dirección física del dato escrito por la instrucción `sw`: `0x_____`
3. Se tiene un procesador MIPS32 con un cache L1 2SWA unificado, con 16 líneas de una palabra de 32 bits (8 conjuntos) y un bit de validez V, bit dirty D, y campo ASID. El caché es virtualmente direccionado. V=0 (inválido), D=0 (no dirty), su política de reemplazo es LRU, su política de escritura es write-back y ante write misses es write-allocate.
- a) Dibujar un diagrama del cache.
b) Sea la dirección `A[31:0]`. ¿Cuáles serían los bits dedicados a índice y cuáles a tag?
1) índice: `A[__:__]`
2) tag: `A[__:__]`
c) Suponer que el CPU realiza una escritura a la dirección `0x5678`. Identificar el conjunto del caché del esquema dibujado que se verificará para ver si los datos correspondientes se encuentran en sus líneas. Si hubiera un cache hit, ¿cuál sería el contenido del tag en la línea de cache correspondiente? ¿Y el bit D luego de ejecutar la escritura?
1) # conjunto verificado: `__`
2) tag ante hit: `0x_____`
3) D luego de la escritura: `__`
d) Estimar el cache hit ratio para el siguiente programa. Asumir que el cache está vacío antes de comenzar la ejecución, y que las pseudo-instrucciones `la` y `li` se implementan con dos instrucciones ALU cada una.

```
.text
.align 5
strlen:
    la    $t0, string
    li    $t1, 0
loop:
    addu  $t2, $t0, $t1
    addu  $t1, $t1, 1
    lb    $t2, 0($t2)
    bne   $t2, $zero, loop

.data
.align 5
string: .asciiz "string"
```

4. Supongamos ejecutar el siguiente programa en una computadora que implementa un cache unificado, no asociativo de 8 líneas de 32 bit de capacidad; y en la cual todas las instrucciones consumen 1 ciclo de clock cuando el sistema de memoria no desacierta. La frecuencia de reloj del procesador es de 1.6 GHz.

```
# Benchmark B1
.aling 5
li    t1, 100
loop: lw    t2, 1028(zero)
      subu  t1, t1, 1
      bnez  t1, loop
```

- a) Calcular la cantidad de accesos a memoria realizados durante la ejecución de este programa.
b) Calcular la cantidad de aciertos, desaciertos, y la tasa de desaciertos.
c) Calcular el CPI efectivo, seleccionando un valor de tiempo de acceso a memoria principal que consideres representativo (en orden de magnitud) para una computadora contemporánea que tenga esa velocidad de clock.