

<b>Comenzado el</b>	lunes, 8 de noviembre de 2021, 19:24
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	sábado, 11 de diciembre de 2021, 15:45
<b>Tiempo empleado</b>	32 días 20 horas
<b>Puntos</b>	Sin calificar aún/20,00

Pregunta **1**

Finalizado

Puntúa como 1,00

Teniendo en cuenta la representación de la secuencia de accesos a memoria de la figura 1, ¿crees que ésta es aleatoria, o por el contrario sigue algún tipo de patrón?

Son predecibles, siguen un patrón

Los accesos presentan un patrón que viene determinado por la localidad de referencias que caracteriza a cualquier programa.

Pregunta **2**

Finalizado

Puntúa como 1,00

Indica un ejemplo de localidad espacial en el acceso al código del programa de conversión a mayúsculas teniendo en cuenta su patrón de accesos a memoria en la figura 1.

El array de String

La localidad espacial en el código se aprecia en la ejecución secuencial de las instrucciones. Serían aquellos accesos que forman una recta con una pendiente de aproximadamente 75° (se incrementan cuatro posiciones de memoria por cada instrucción).

Hay que tener en cuenta que en la arquitectura MIPS64 el código de las instrucciones son de 32 bits - 4 bytes - y se accede a ellos en una única operación, con lo cual se está accediendo simultáneamente a las posiciones:  $n$ ,  $n+1$ ,  $n+2$  y  $n+3$ .

Pregunta **3**

Finalizado

Puntúa como 1,00

Indica un ejemplo de localidad temporal en el acceso al código del programa de conversión a mayúsculas teniendo en cuenta su patrón de accesos a memoria en la figura 1.

Bucle loop

La localidad temporal en el código se aprecia en la ejecución de bucles. Esto se observa como patrones idénticos (mismas direcciones) que se repiten en el tiempo.

Pregunta **4**

Finalizado

Puntúa como 1,00

Indica un ejemplo de localidad espacial en el acceso a los datos para el programa de conversión a mayúsculas teniendo en cuenta el patrón de accesos de la figura 1.

Cuando se escribe

La localidad espacial en los datos se produce cuando se recorren estructuras de datos secuencialmente (arrays). En este caso se trataría de los accesos a la cadena de texto.

Pregunta **5**

Sin contestar

Puntúa como 1,00

Indica un ejemplo de localidad temporal en el acceso a los datos para el programa de conversión a mayúsculas teniendo en cuenta el patrón de accesos de la figura 1.


La localidad temporal en los datos se observa en el acceso a variables de control de bucles o acceso repetido a posiciones de arrays. En este caso se trataría de los accesos repetidos a las posiciones de la cadena (uno para leer y otro para modificar el carácter).

Pregunta **6**

Correcta

Puntúa 1,00 sobre 1,00

¿Cuántos bloques de memoria contiene la memoria principal del sistema a diseñar?

Respuesta:  

4096 bloques

$64\text{Kpalabras} / 16 \text{ palabras por bloque} = 2^{16}/2^4 = 2^{12} = 4096$

La respuesta correcta es: 4096

Pregunta **7**

Correcta

Puntúa 1,00 sobre 1,00

¿Cuál es tamaño de la memoria caché expresado en bytes del sistema diseñar?

Respuesta:  

64 bytes

$4 \text{ bloques} * 16 \text{ palabras/bloque} * 1 \text{ byte/palabra} = 2^2 * 2^4 * 1 = 2^6 = 64$


La respuesta correcta es: 64

Pregunta **8**

Correcta

Puntúa 1,00 sobre 1,00

A partir de la información de la traza, ¿cuántos accesos a memoria se realizan durante la ejecución del programa de conversión a mayúsculas?

Respuesta:  

125 accesos en la traza del programa

La respuesta correcta es: 125

Pregunta **9**

Incorrecta

Puntúa 0,00 sobre 1,00

Si solo existiera el nivel de memoria principal, ¿cuántos ciclos de reloj consumirían los accesos a memoria del programa de conversión a mayúsculas?

Respuesta:  ❌

2500 ciclos de reloj

Nº de accesos \* Tpo en ciclos de cada acceso =  $125 * 20 = 2500$

La respuesta correcta es: 2500

Pregunta **10**

Correcta

Puntúa 1,00 sobre 1,00

Teniendo en cuenta la frecuencia de reloj, ¿cuánto tiempo consumen los accesos a memoria del programa de conversión a mayúsculas?

Respuesta:  ✅

2500 nanosegundos

En la pregunta anterior se calculó el número de ciclos de reloj ( $125 * 20 = 2500$  ciclos de reloj)

Para obtener el tiempo se multiplica el número de ciclos por el periodo de reloj (inversa de la frecuencia). En este caso la frecuencia es de 1 GHz, o lo que es lo mismo  $1 \times 10^9$ . Por tanto el periodo será de  $1/(1 \times 10^9)$  o  $1 \times 10^{-9}$  segundos.

El tiempo de acceso será de  $2500 \times 10^{-9}$  segundos, o teniendo en cuenta que  $10^{-9}$  es un nanosegundo, el tiempo podría expresarse también como 2500 nanosegundos.

La respuesta correcta es: 2500 ns

Pregunta **11**

Correcta

Puntúa 1,00 sobre 1,00

¿Cuál es la tasa de aciertos para el sistema de memoria y la traza de accesos a memoria del programa de conversión a mayúsculas? Escribe el resultado en tanto por ciento sin símbolo final.

Respuesta:  ✅

97,6 %

La tasa de aciertos se obtiene dividiendo el número de aciertos en memoria caché entre el número total de accesos ( $122/125$ ).

La respuesta correcta es: 97,6

Pregunta **12**

Sin contestar

Puntúa como 1,00

Teniendo en cuenta la tasa de aciertos obtenida en la pregunta anterior, así como el número total de accesos, ¿cuál será el número de ciclos consumidos para realizar todos los accesos a memoria del programa de conversión a mayúsculas?

$34 \text{ fallos} \times 16/4 \text{ lecturas} \times 20 \text{ ciclos/lectura} + 122 \text{ aciertos} \times 1 \text{ ciclo} = 362 \text{ ciclos}$

Se desprecia el tiempo de acceso a la caché frente al tiempo necesario para mover un bloque de memoria principal a caché cuando hay fallo de caché. Debe tenerse en cuenta que para satisfacer un fallo de caché hay que leer 16 bytes de memoria principal. Como en cada operación de lectura se leen 4 bytes a la vez, hacen falta 16/4 lecturas para mover un bloque de memoria principal a caché.

Pregunta **13**

Incorrecta

Puntúa 0,00 sobre 1,00

Con la jerarquía de memoria desarrollada, ¿cuál es el tiempo necesario para completar los accesos a memoria del programa de conversión a mayúsculas?

Respuesta:  ❌

En la pregunta anterior se calculo el número total de ciclos que requerían los accesos:

- 122 ciclos para cada acierto en cache

-  $16/4 \times 20 \times 3 = 240$  para los cuatro fallos de caché

En total 362 ciclos.

El tiempo empleado se calcularía multiplicando el número de ciclos por el tiempo por ciclo (inverso de su frecuencia)

Como el periodo de reloj de la CPU es de 1 nanosegundo ( $1 \times 10^{-9}$  segundos), el tiempo empleado será de  $362 \times \text{Tiempo por ciclo}$ , en este caso 362 nanosegundos.

La respuesta correcta es: 362

Pregunta **14**

Incorrecta

Puntúa 0,00 sobre 1,00

Suponiendo que el patrón de accesos a memoria fuese totalmente aleatorio para el programa de conversión a mayúsculas (se considera que no tendría ningún tipo de localidad). ¿Cuál sería la tasa de aciertos de caché empleando la jerarquía formada por la memoria principal y la caché? Responder en porcentaje sin el símbolo final.

Respuesta:  ❌

$4/4096 = 0,00097$  ó  $0,097\%$

Sería la probabilidad de que se ocuparan los 4 bloques en caché entre los 4096 bloques que tiene la memoria principal


La respuesta correcta es: 0,00097

Pregunta **15**

Incorrecta

Puntúa 0,00 sobre 1,00

Suponiendo el caso anterior en el que se considera un patrón de accesos totalmente aleatorio, ¿cuál sería en este caso el tiempo medio de acceso a memoria?

Respuesta:  

La expresión general del tiempo medio de acceso para esta jerarquía sería:

$\text{Acaché} * \text{Tpo. caché} + (1 - \text{Acaché}) * \text{Tpo. Mem} * \text{N}^\circ \text{ accesos a memoria}$

tasa de acierto en cache (Acaché) =  $4/4096$ ;

Tpo. acceso a cache = 1 ns;

Tpo. acceso a memoria principal = 20 ns;

Tamaño del bloque = 16;

$4/4096 * 1 + (1 - (4/4096)) * 20 * (16/4) = 80 \text{ ns}$


La respuesta correcta es: 320

Pregunta **16**

Sin contestar

Puntúa como 1,00

En las mismas condiciones de patrón de acceso aleatorio y teniendo en cuenta la respuesta de la pregunta anterior, ¿cuál sería el tiempo necesario para completar todos los accesos a memoria?

Respuesta:  

10000 nanosegundos

tiempo total = tiempo acceso medio \* número accesos;

$80 \text{ nanosegundos} * 125 = 10000 \text{ nanosegundos}$

La respuesta correcta es: 10000

Pregunta **17**

Sin contestar

Puntúa como 1,00

En el caso de tener el patrón de accesos aleatorio, compara el tiempo medio de acceso obtenido utilizando memoria caché con el tiempo medio de acceso a la memoria principal. ¿Merece la pena el empleo de la jerarquía de memoria, o lo que es lo mismo, el empleo de la caché? ¿Por qué?

Tiempo medio de acceso utilizando la jerarquía = 80 nanosegundos, calculado en la pregunta 16.

Tiempo medio de acceso utilizando sólo la memoria principal = 20 nanosegundos

No merece la pena usar jerarquía porque el acceso es incluso más costoso con la caché que sin ella

Pregunta **18**

Sin contestar

Puntúa como 1,00

Suponiendo el caso ideal en el cual la localidad fuese tan elevada, o la caché tan grande, que se consiguiese una tasa de aciertos de caché del 100%, ¿cuál sería el tiempo medio de acceso a memoria?

Respuesta:  

1 nanosegundo

Todos los accesos se servirían desde la memoria caché, ese sería por tanto el tiempo de acceso medio.

La respuesta correcta es: 1 ns

Pregunta **19**

Sin contestar

Puntúa como 1,00

En el caso ideal, visto en la pregunta anterior, ¿qué particularidad tiene el tiempo de acceso medio obtenido?

Coincide con el tiempo de acceso a la memoria caché, ya que todos los accesos se sirven desde la memoria caché.

Pregunta **20**

Sin contestar

Puntúa como 1,00

¿Cuál sería en este caso ideal el tiempo necesario para completar todos los accesos a memoria?

Respuesta:



125 ns

Número de accesos \* tiempo de acceso a memoria caché

La respuesta correcta es: 125

[◀ Sesión 2.4 - Soporte a los SO multitarea y a la virtualización](#)

Ir a...



[Sesión 3.2 -Prueba de estrategias de correspondencia de memoria caché ▶](#)