|  |
| --- |
| **Assignment # 5**  **SYSC 5704 – Elements of Computer Systems** |
| Fall 2014  Submitted To  Dr. R. Gregory Franks  By  **Ferhan Jamal (100 953 487)**  Carleton University |

**5.1 In this exercise we look at memory locality properties of matrix computation. The following code is written in C, where elements within the same row are stored contiguously. Assume each word is a 32-bit integer.**

**for (I=0; I<8; I++)**

**for (j=0; J<8000; J++)**

**A[I][J] = B[I][0]+A[J][I];**

**1. [5] <§5.1> How many 32-bit integers can be stored in a 16-byte cache block?**

The 16-byte cache block can store 16\*8bits=128 bits and 1 integer is equal to 32 bits so the number of integers in the cache block will be: 128/32=4

**2. [5] <§5.1> References to which variables exhibit temporal locality?**

The temporal locality is based on the concept that a resource or memory location which is referenced at one point in time will be referenced in the near future too. The references to which variables exhibit temporal locality are:-

I, J, and B[I][0]

**3. [5] <§5.1> References to which variables exhibit spatial locality?**

The spatial locality is based on the concept that if a particular memory location or resource is referenced at one point of time then it is likely that nearby memory locations or resource will be referenced in the near future too. The references to which variables exhibit spatial locality are:-

A[I][J]

**Locality is affected by both the reference order and data layout. The same computation can be written below in Matlab, which differs from from C by storing matrix elements from the same column contiguously in memory.**

**for I=I:8**

**for J=1:8000**

**A(I,J)=B(I,0)+A(J,I)**

**end**

**end**

**4. [10] <§5.1> How many 16-byte cache blocks are needed to store all 32-bit matrix elements being referenced?**

In the loops,

64000(8\*8000) elements is referenced by A(I,J)

8(8\*1) elements is referenced by B(I,0)

64000(8\*8000) elements is referenced by A(J,I)

Also, A(I, J) and A(J, I) references 64 elements in common for values of I, J <8 which implies that the total number of elements referenced is

64,000\*2 + 8 – 64 = 1,27,944 elements.

Each element among 1,27,944 elements is a 32-bit integer i.e. 4 bytes which implies that 1,27,944 \*4= 5,11,776 bytes

Finally, we can say that to store all the referenced matrix elements in the cache, we need

511,776/16= 31,986 blocks in the cache

**5. [10] <§5.1> References to which variables exhibit temporal locality?**

The temporal locality is based on the concept that a resource or memory location which is referenced at one point in time will be referenced in the near future too. The references to which variables exhibit temporal locality are:-

I, J, and B(I,0)

**6. [10] <§5.1> References to which variables exhibit spatial locality?**

The spatial locality is based on the concept that if a particular memory location or resource is referenced at one point of time then it is likely that nearby memory locations or resource will be referenced in the near future too. The references to which variables exhibit spatial locality are:-

A[J][I]

**5.5 Media applications that play audio or video files are part of a class of workloads called “streaming” workloads; i.e., they bring in large amounts of data but do not reuse much of it. Consider a video streaming workload that accesses a 512 KiB working set sequentially with the following address stream:**

**0, 2, 4, 6, 8, 10, 12, 14, 16, ...**

**1. [5] <§§5.4,5.8> Assume a 64 KiB direct-mapped cache with a 32-byte. What is the miss rate for the address stream above? How is this miss rate sensitive to the size of the cache or the working set? How would you categorize the misses this workload is experiencing, based on the 3C model?**

It is given in the question that a video streaming workload that accesses a 512 KiB working set sequentially with the following address stream:

0, 2, 4, 6, 8, 10, 12, 14, 16, ...

In a 64KiB direct-mapped cache with a 32-byte, every 8th lookup will be a miss which further implies that the miss rate will be: 1 / 4=25%.

Note: 64KiB cache has 64\*1024/32=1024 blocks. Accessing the 32-byte cache blocks, there will be 1 miss and 7 hits from the starting blocks and so on.

This miss rate is totally insensitive to the size of the cache and the size of the working set.

The misses that this workload is experiencing, based on the 3C model can be categorized as cold start-misses or compulsory misses.

**2. [5] <§§5.1, 5.8> Re-compute the miss rate when the cache block size is 16 bytes, 64 bytes, and 128 bytes. What kind of locality is this workload exploiting?**

**In case of 16-bytes, the miss rate will be:**

1/4 = 25% (Every 4th look-up is a miss)

**In case of 64-bytes, the miss rate will be:**

1/16 = 6.25% (Every 16th look-up is a miss)

**In case of 128-bytes, the miss rate will be:**

1/32 = 3.125% (Every 32th look-up is a miss)

This workload is exploiting spatial locality.

**3. [10] <§5.13> “Prefetching” is a technique that leverages predictable address patterns to specttlatively bring in additional cache blocks when a particular cache block is accessed. One example of prefetching is a stream buffer that prefetches sequentially adjacent cache blocks into a separate buffer when a particular cache block is brought in. If the data is found in the prefetch buffer, it is considered as a hit and moved into the cache and the next cache block is prefetched. Assume a two-entry stream buffer and assume that the cache latency is such that a cache block can be loaded before the computation on the previous cache block is completed. What is the miss rate for the address stream above?**

It is given in the question, that assume a 2-entry stream buffer and assume that the cache latency is such that a cache block can be loaded before the computation on the previous cache block is completed. t is also mentioned in the question that stream buffer uses prefetching example that prefetches adjacent sequential blocks into a separate buffer when a particular cache block is brought in.

One example of prefetching is a stream buffer that prefetches sequentially adjacent cache blocks into a separate buffer when a particular cache block is brought in. It is also mentioned in the question that stream buffer uses prefetching example that prefetches adjacent sequential blocks into a separate buffer when a particular cache block is brought in.

Adjacent sequential blocks are prefetched into a separate buffer when a particular cache block is brought in so the miss rate will be 0% since all blocks will be available in the pre-fetch buffer and there will be total hits.

The miss-rate will be 0%.

**Cache block size (B) can affect both miss rate and miss latency. Assuming a 1-CPI machine with an average of 1.35 references (both instruction and data) per instruction, help find the optimal block size given the following miss rates for various block sizes.**

**8: 4% 16: 3% 32: 2% 64: 1.5% 128: 1%**

**4. [10] <§5.3> What is the optimal block size for a miss latency of 20 × B cycles?**

It is given in the question that there is 1-CPI machine with an average of 1.35 references (both instruction and data) per instruction. Now, we have to find the optimal block size for a miss latency of 20 x B cycles.

|  |  |  |  |
| --- | --- | --- | --- |
| Block Size(B)  bytes | Miss Rate (%) | Latency (20XB)  cycles | Effective CPI |
| 8 | 4 | 160 | 9.6 |
| 16 | 3 | 320 | 13.93 |
| 32 | 2 | 640 | 18.26 |
| 64 | 1.5 | 1280 | 26.905 |
| 128 | 1 | 2560 | 35.55 |

The optimal block size here is 8 bytes.

**5. [10] <§5.3> What is the optimal block size for a miss latency of 24 + B cycles?**

In the case of 24+B cycles, we have:

|  |  |  |  |
| --- | --- | --- | --- |
| Block Size(B)  bytes | Miss Rate (%) | Latency (24+B)  Cycles | Effective CPI |
| 8 | 4 | 32 | 2.68 |
| 16 | 3 | 40 | 2.59 |
| 32 | 2 | 56 | 2.49 |
| 64 | 1.5 | 88 | 2.76 |
| 128 | 1 | 152 | 3.04 |

(0.96 \* 1) + (1.35 \* 0.04 \* 32)= 2.68

(0.97 \* 1) + (1.35 \* 0.03 \* 40)= 2.59

(0.98 \* 1) + (1.35 \* 0.02 \* 56)= 2.492

(0.985 \* 1) + (1.35 \* 0.015 \* 88)= 2.76

(0.99 \* 1) + (1.35 \* 0.01 \* 152)= 3.04

The optimal block size is 32-byte block here.

**6. [10] <§5.3> For constant miss latency, what is the optimal block size?**

For a constant miss latency, the optimal block size is the 128-byte block because it has the lowest miss rate and the largest block size.

**5.19 In this exercise we show the definition of a web server log and examine code optimizations to improve log processing speed. The data structure for the log is defined**

**struct entry {**

**int srcIP; // remote IP address**

**char URL[128]; // request URL (e.g.. "GET index.html")**

**long long refTime; // reference time**

**int status; // connection status**

**char browser[64]; // client browser name**

**} log [NUM\_ENTRIES];**

**Assume the following processing function for the log:**

**topK\_sourceIP (int hour);**

**This function finds the top 1000 IP addresses during the period specified.**

**1. [5] <§5.15> Which fields in a log entry will be accessed for the given log processing function? Assuming 64-byte cache blocks and no prefetching, how many cache misses per entry does the given function incur on average?**

The above data structure for the log is 40 bytes long. The individual split-ups are as follows:-

srcIP: 4 bytes

URL: 16 bytes

refTime: 8 bytes

status: 4 bytes

browser: 8 bytes

Total: 40 bytes

In the 1st part of the question we have to find the fields in a log entry which will be accessed for the given log processing function. The function topK\_sourceIP(int hour) will reference the following 2 fields:-

a. srcIP field (It will retrieve the IP address)

b. refTime field (It will check whether it is falling in the given hour time frame)

There will be 1 cache miss per entry the given function incur on average with every entry is only accessed once because there will be only one srcIP along with its specific refTime in a block of cache at a given instance of time with no prefetching in a given particular order of the data structure.

**2. [10] <§5.15> How can you reorganize the data structure to improve cache utilization and access locality? Show your structure definition code.**

Below are the ways we can recognize the data structure to improve cache utilization and access locality:-

a. The 1st way is to move the refTime field towards the beginning of the struct as:

struct entry{

int srcIP; // remote IP address

long long refTime; // reference time

char URL[128]; // request URL

int status; // connection status

char browser[64]; // client browser name

} log [NUM\_ENTRIES];

b. The 2nd possible is to shrink or reduce the struct itself. For example:- If the URL fields (for example) are not being used then remove it from the struct completely.

struct entry{

int srcIP; // remote IP address

long long refTime; // reference time

int status; // connection status

char browser[64]; // client browser name

} log [NUM\_ENTRIES];

Both of the above ways will make the structure size more compatible and will more entries into the cache and will improve the cache utilization and access locality.

**3. [10] <§5.15> Another example of a log processing function finds the peak hours that connection request has the given status, e.g.:**

**peak\_hour (int status); // peak hours of a given status**

**If both functions are important, how would you reorganize the data structure to improve the overall performance?**

We have 2 choices if we want to improve the overall performance of the function and the log function together. Both of the choices are:-

a. The 1st choice is, we can remove the fields (char arrays) from the struct to make it smaller.

struct entry{

int srcIP; // remote IP address

long long refTime; // reference time

int status; // connection status

} log [NUM\_ENTRIES];

b. The 2nd choice is that if we wish not to remove the URL and browser in the structure then we can use pointers in the struct to those fields. In this choice there is no need to eliminate the fields.

Both of these choices will make the size more compatible and will improve the overall performance of the data structure.

**5.11**

It is given in the question that the address stream has the values:

4669, 2227, 13916, 34587, 48870 12608 49225

|  |  |  |
| --- | --- | --- |
| TLB | | |
| Valid | Tag | Physical Page Number |
| 1 | 11 | 12 |
| 1 | 7 | 4 |
| 1 | 3 | 6 |
| 0 | 4 | 9 |

|  |  |
| --- | --- |
| Page Table | |
| Valid | Physical Page or in Disk |
| 1 | 5 |
| 0 | disk |
| 0 | disk |
| 1 | 6 |
| 1 | 9 |
| 1 | 11 |
| 0 | disk |
| 1 | 4 |
| 0 | disk |
| 0 | disk |
| 1 | 3 |
| 1 | 12 |
| 0 | disk |

1. **[10] <§5.7>**

For the given address stream using 4 KiB pages and with the initial TLB and page table, we have to show the final state of the system. We have now:

|  |  |
| --- | --- |
| Address | Virtual Page (Tag) |
| 4669 | 1 |
| 2227 | 0 |
| 13916 | 3 |
| 34587 | 8 |
| 48870 | 11 |
| 12608 | 3 |
| 49225 | 12 |

After referencing 4669, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 4669 | 1 | Hit/Miss | Valid | Tag | Physical Page |
| Page Fault | 1 | 11 | 12 |
| 1 | 7 | 4 |
| 1 | 3 | 6 |
| 1(last access 0) | 1 | 13 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing 2227, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 2227 | 0 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Miss  PT Hit | 1(last access 0) | 0 | 5 |
| 1 | 7 | 4 |
| 1 | 3 | 6 |
| 1(last access 1) | 1 | 13 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing 13916, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 13916 | 3 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Hit | 1(last access 1) | 0 | 5 |
| 1 | 7 | 4 |
| 1(last access 0) | 3 | 6 |
| 1(last access 2) | 1 | 13 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 34587, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 34587 | 8 | Hit/Miss | Valid | Tag | Physical Page |
| Page Fault | 1(last access 2) | 0 | 5 |
| 1(last access 0) | 8 | 4 |
| 1(last access 1) | 3 | 6 |
| 1(last access 3) | 1 | 13 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 1 | 14 |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 48870, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 48870 | 11 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Miss  PT Hit | 1(last access 3) | 0 | 5 |
| 1(last access 1) | 8 | 14 |
| 1(last access 2) | 3 | 6 |
| 1(last access 0) | 11 | 12 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 12608, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 12608 | 3 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Miss  PT Hit | 1(last access 4) | 0 | 5 |
| 1(last access 2) | 8 | 14 |
| 1(last access 0) | 3 | 6 |
| 1(last access 1) | 11 | 12 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | Disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | Disk |
| 7 | 1 | 4 |
| 8 | 0 | Disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | Disk |

After referencing address 49225, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 49225 | 12 | Hit/Miss | Valid | Tag | Physical Page |
| Page Fault | 1(last access 0) | 12 | 15 |
| 1(last access 3) | 8 | 14 |
| 1(last access 1) | 3 | 6 |
| 1(last access 2) | 11 | 12 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 1 | 15 |

That’s how, the final state of the system will be.

**2. [15] <§5.7>**

In this part of the question for the given address stream we have to use 16 KiB pages instead of 4KiB pages, we have:

|  |  |
| --- | --- |
| **Address** | **Virtual Page(Tag) (((Tag)** |
| 4669 | 0 |
| 2227 | 0 |
| 13916 | 0 |
| 34587 | 2 |
| 48870 | 2 |
| 12608 | 0 |
| 49225 | 3 |

After referencing address 4669, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 4669 | 0 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Miss  PT Hit | 1 | 11 | 12 |
| 1 | 7 | 4 |
| 1 | 3 | 6 |
| 1(last access 0) | 0 | 5 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 2227, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 2227 | 0 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Hit | 1 | 11 | 12 |
| 1 | 7 | 4 |
| 1 | 3 | 6 |
| 1(last access 0) | 0 | 5 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 13916, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 13916 | 0 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Hit | 1 | 11 | 12 |
| 1 | 7 | 4 |
| 1 | 3 | 6 |
| 1(last access 0) | 0 | 5 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 34587,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 34587 | 2 | Hit/Miss | Valid | Tag | Physical Page |
| Page Fault | 1(last access 0) | 2 | 13 |
| 1 | 7 | 4 |
| 1 | 3 | 6 |
| 1(last access 1) | 0 | 5 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 0 | disk |
| 2 | 1 | 13 |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 48870, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 48870 | 2 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Hit | 1(last access 0) | 2 | 13 |
| 1 | 7 | 4 |
| 1 | 3 | 6 |
| 1(last access 2) | 0 | 5 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 12608, we have:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 12608 | 0 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Hit | 1(last access 1) | 2 | 13 |
| 1 | 7 | 4 |
| 1 | 3 | 6 |
| 1(last access 0) | 0 | 5 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 49225,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | TLB | | | |
| 49225 | 3 | Hit/Miss | Valid | Tag | Physical Page |
| TLB Hit | 1(last access 2) | 2 | 13 |
| 1 | 7 | 4 |
| 1(last access 0) | 3 | 6 |
| 1(last access 1) | 0 | 5 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

The final state of the system will be like this.

**3. [15] <§§5.4, 5.7>**

For the 2-way set associative TLB, we have:

After referencing address 4669,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 4669 | 1 | 0 | 1 | Hit/Miss | Valid | Tag | Physical Page | Index |
| Page Fault | 1 | 11 | 12 | 0 |
| 1 | 7 | 4 | 1 |
| 1 | 3 | 6 | 0 |
| 1 (last access 0) | 0 | 13 | 1 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 2227,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 2227 | 0 | 0 | 0 | Hit/Miss | Valid | Tag | Physical Page | Index |
| TLB Miss PT Hit | 1 (last access 0) | 0 | 5 | 0 |
| 1 | 7 | 4 | 1 |
| 1 | 3 | 6 | 0 |
| 1 (last access 1) | 0 | 13 | 1 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 0 | 1 | 5 |

After referencing address 13916, we have:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 13916 | 3 | 1 | 1 | Hit/Miss | Valid | Tag | Physical Page | Index |
| TLB Miss PT Hit | 1 (last access 1) | 0 | 5 | 0 |
| 1 (last access 0) | 1 | 6 | 1 |
| 1 | 3 | 6 | 0 |
| 1 (last access 2) | 0 | 13 | 1 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 34587,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 34587 | 8 | 4 | 0 | Hit/Miss | Valid | Tag | Physical Page | Index |
| Page Fault | 1 (last access 2) | 0 | 5 | 0 |
| 1 (last access 1) | 1 | 6 | 1 |
| 1 (last access 0) | 4 | 14 | 0 |
| 1 (last access 3) | 0 | 13 | 1 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 1 | 14 |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 48870, we have:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 48870 | 11 | 5 | 1 | Hit/Miss | Valid | Tag | Physical Page | Index |
| TLB Miss PT Hit | 1 (last access 3) | 0 | 5 | 0 |
| 1 (last access 2) | 1 | 6 | 1 |
| 1 (last access 1) | 4 | 14 | 0 |
| 1 (last access 0) | 5 | 12 | 1 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 1 | 14 |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 12608, we have:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 12608 | 3 | 1 | 1 | Hit/Miss | Valid | Tag | Physical Page | Index |
| TLB Hit | 1 (last access 4) | 0 | 5 | 0 |
| 1 (last access 0) | 1 | 6 | 1 |
| 1 (last access 2) | 4 | 14 | 0 |
| 1 (last access 1) | 5 | 12 | 1 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 1 | 14 |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 49225, we have:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 49225 | 12 | 6 | 0 | Hit/Miss | Valid | Tag | Physical Page | Index |
| Page Fault | 1 (last access 0) | 6 | 15 | 0 |
| 1 (last access 1) | 1 | 6 | 1 |
| 1 (last access 3) | 4 | 14 | 0 |
| 1 (last access 2) | 5 | 12 | 1 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 1 | 14 |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 1 | 15 |

This will be the final contents of the the TLB if it is a 2-way set associative.

For the direct-mapped TLB, we have:

After referencing address 4669,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 4669 | 1 | 0 | 1 | Hit/Miss | Valid | Tag | Physical Page | Index |
| Page Fault | 1 | 11 | 12 | 0 |
| 1 | 0 | 13 | 1 |
| 1 | 3 | 6 | 2 |
| 0 | 4 | 9 | 3 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 2227,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 2227 | 0 | 0 | 0 | Hit/Miss | Valid | Tag | Physical Page | Index |
| TLB Miss  PT Hit | 1 | 0 | 5 | 0 |
| 1 | 0 | 13 | 1 |
| 1 | 3 | 6 | 2 |
| 0 | 4 | 9 | 3 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 13916,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 13916 | 3 | 0 | 3 | Hit/Miss | Valid | Tag | Physical Page | Index |
| TLB Miss  PT Hit | 1 | 0 | 5 | 0 |
| 1 | 0 | 13 | 1 |
| 1 | 3 | 6 | 2 |
| 1 | 0 | 6 | 3 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 0 | disk |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 34587, we have:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 34587 | 8 | 2 | 0 | Hit/Miss | Valid | Tag | Physical Page | Index |
| Page Fault | 1 | 2 | 14 | 0 |
| 1 | 0 | 13 | 1 |
| 1 | 3 | 6 | 2 |
| 1 | 0 | 6 | 3 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 1 | 14 |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 48870,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 48870 | 11 | 2 | 3 |  | Valid | Tag | Physical Page | Index |
| TLB Miss  PT Hit | 1 | 2 | 14 | 0 |
| 1 | 0 | 13 | 1 |
| 1 | 3 | 6 | 2 |
| 1 | 2 | 12 | 3 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 1 | 14 |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 12608,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 12608 | 3 | 0 | 3 | Hit/Miss | Valid | Tag | Physical Page | Index |
| TLB Miss  PT Hit | 1 | 2 | 14 | 0 |
| 1 | 0 | 13 | 1 |
| 1 | 3 | 6 | 2 |
| 1 | 0 | 6 | 3 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 1 | 14 |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 0 | disk |

After referencing address 49225, we have:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Address | Virtual Page | Tag | Index | TLB | | | | |
| 499225 | 12 | 3 | 0 | Hit/Miss | Valid | Tag | Physical Page | Index |
| Page Fault | 1 | 3 | 15 | 0 |
| 1 | 0 | 13 | 1 |
| 1 | 3 | 6 | 2 |
| 1 | 0 | 6 | 3 |

|  |  |  |
| --- | --- | --- |
| Page Table | | |
| Tag | Valid | Physical Page or in Disk |
| 0 | 1 | 5 |
| 1 | 1 | 13 |
| 2 | 0 | disk |
| 3 | 1 | 6 |
| 4 | 1 | 9 |
| 5 | 1 | 11 |
| 6 | 0 | disk |
| 7 | 1 | 4 |
| 8 | 1 | 14 |
| 9 | 0 | disk |
| 10 | 1 | 3 |
| 11 | 1 | 12 |
| 12 | 1 | 15 |

This will be the final state of the TLB in case direct-mapped TLB.

**4. [5] <§5.7>**

Number of page table entries required per process will be

= 262,144 entries [ 2(31-13)  = 218 ]

As each page table entry size is 4 bytes, therefore we have:

262,144 \* 4 = 1,048,576 bytes = 1024 KiB = 1 MiB which implies that every process requires 1 MiB for its page table, therefore for 5 processes, we have:

= 5 MiB

**5. [10] <§5.7>**

Number of total leaf page table entries required per process will be

2(31-13) = 218  = 262,144 entries

As we have a 2-level page table approach with 256 entries in the 1st level then each of the 2nd level page table contains,

262,144/256 = 1024 entries

Each of the entries are of 4-bytes, therefore we have:-

1024 \*4= 4096 bytes.

Each of the 1st level page table has 256 entries each of 6 bytes, therefore,

256\*6= 1536 bytes.

The minimum amount of memory required :

4096+1536 = 5.5 KiB

The maximum amount of memory required (including 1st page and 2nd level page tables,) i.e.

1536 + (4096 \* 256) = 1025.5 KiB (1,050,112 bytes)