



**Nanyang Technological University**  
**School of Computer Science and Engineering**

---

## **CZ2005 OPERATING SYSTEMS LAB 3 REPORT**

**Name:**

**LEE SUYEON**

**Matric ID:**

**N1801612A**

**Group Number:**

**SSR3**

**Submission Date:**

**8, April, 2019**

# 1. Code Implementation

## (1) int VpnToPhyPage(int vpn)

: Gets a physical frame phyPage for a virtual page vpn, if exists in the IPT.

```
38 //-----
39 // VpnToPhyPage
40 //     Gets a phyPage for a vpn, if exists in ipt.
41 //-----
42
43 int VpnToPhyPage(int vpn)
44 {
45     //your code here to get a physical frame for page vpn
46     int i = 0;
47
48     for(; i < NumPhysPages; i++){
49         if(memoryTable[i].valid == true && memoryTable[i].pid == currentThread->pid && memoryTable[i].vPage == vpn){
50             return i;
51         }
52     }
53
54     return -1; //not found
55     //you can refer to PageOutPageIn(int vpn) to see how an entry was created in ipt
56 }
```

## (2) void InsertToTLB(int vpn, int phyPage)

: Insert a vpn / phyPage entry into the TLB.

```
58 //-----
59 // InsertToTLB
60 //     Put a vpn/phyPage combination into the TLB. If TLB is full, use FIFO
61 //     replacement
62 //-----
63
64 static int FIFO = 0;
65 void InsertToTLB(int vpn, int phyPage)
66 {
67     int i = 0; //entry in the TLB
68     //your code to find an empty in TLB or to replace the oldest entry if TLB is full
69     for(; i < TLBSize; i++){
70         if(machine->tlb[i].valid == false){
71             DEBUG('p', "This is invalid entry");
72             break;
73         }
74     }
75
76     if(i == TLBSize){
77         i = FIFO;
78     }
79
80     FIFO = (i + 1) % TLBSize;
```

(there are more code segments below for this function, but omitted in the report)

The replacement policy of the TLB is FIFO (First In First Out).

### (3) int lruAlgorithm(void)

: Return the freed physical frame according to the least recently used algorithm.

```
220  //-----
221  // lruAlgorithm
222  // Determine where a vpn should go in phymem, and therefore what
223  // should be paged out. This lru algorithm is the one discussed in the
224  // lectures.
225  //-----
226
227  int lruAlgorithm(void)
228  {
229      //your code here to find the physical frame that should be freed
230      //according to the LRU algorithm.
231      int phyPage;
232      int i = 0;
233      for(; i < NumPhysPages; i++){
234          if(memoryTable[i].valid == false){
235              phyPage = i;
236              break;
237          }
238      } //if there is invalid one, return it
239
240      if(i == NumPhysPages){
241          int min = memoryTable[0].lastUsed;
242          int min_index = 0;
243          for(i = 0; i < NumPhysPages; i++){
244              if(memoryTable[i].lastUsed < min){
245                  min = memoryTable[i].lastUsed;
246                  min_index = i;
247              }
248          }
249          phyPage = min_index;
250      }
251
252      return phyPage;
253  }
```

## 2. How program works

- 1) When pid and vpn given, first search TLB if there is corresponding TLBentry.
- 2-1) If there is corresponding entry, it is TLB hit. We can get physical address.
- 2) If there is no corresponding entry, it is TLB miss. We have to search for IPT in main memory.
- 3-1) If there is searching vpn in IPT slot, we do not need to do paging in.

3) If there is no searching vpn in IPT slot, we need to do paging in by proper phyPage value. If there is IPT slot which has invalid bit, return the index of the IPT slot. If all IPT slots are valid, choose the page according to least recently used (lru) algorithm and return its slot index. Returned value becomes phyPage.

4) Insert corresponding TLBentry for current vpn. The replacement policy of the TLB is FIFO (First In First Out).

### 3. Test Program Output

tick	vpn	pid	IPT[0] pid,vpn, lastUsed, valid	IPT[1]	IPT[2]	IPT[3]	TLB[0] vpn, phy, valid	TLB[1]	TLB[2]	Page Out
10	0	0	0,0,0,0	0,0,0,0	0,0,0,0	0,0,0,0	0,0,0	0,0,0	0,0,0	
13	9	0	0,0,12,1	0,0,0,0	0,0,0,0	0,0,0,0	0,0,1	0,0,0	0,0,0	
15	26	0	0,0,12,1	0,9,15,1	0,0,0,0	0,0,0,0	0,0,1	9,1,1	0,0,0	
20	1	0	0,0,12,1	0,9,19,1	0,26,17,1	0,0,0,0	0,0,1	9,1,1	26,2,1	
26	0	0	0,0,12,1	0,9,25,1	0,26,17,1	0,1,22,1	1,3,1	9,1,1	26,2,1	
28	10	0	0,0,28,1	0,9,25,1	0,26,17,1	0,1,22,1	1,3,1	0,0,1	26,2,0	2
41	9	0	0,0,40,1	0,9,25,1	0,10,28,1	0,1,22,1	1,3,1	0,0,1	10,2,1	
42	26	0	0,0,40,1	0,9,42,1	0,10,28,1	0,1,22,1	9,1,1	0,0,1	10,2,1	
47	0	0	0,0,40,1	0,9,46,1	0,10,28,1	0,26,44,1	9,1,1	26,3,1	10,2,1	
59	0	1	0,0,49,1	0,9,46,1	0,10,28,1	0,26,44,1	9,1,0	26,3,0	0,0,0	
62	9	1	0,0,49,1	0,9,46,1	1,0,61,1	0,26,44,1	0,2,1	26,3,0	0,0,0	3
64	26	1	0,0,49,1	0,9,46,1	1,0,61,1	1,9,64,1	0,2,1	9,3,1	0,0,0	
69	1	1	0,0,49,1	1,26,66,1	1,0,61,1	1,9,68,1	0,2,1	9,3,1	26,1,1	
74	0	1	1,1,71,1	1,26,66,1	1,0,61,1	1,9,73,1	1,0,1	9,3,1	26,1,1	
117	0	0	1,1,71,0	1,26,66,0	1,0,76,0	1,9,73,0	1,0,0	0,2,0	26,1,0	
120	9	0	0,0,119,1	1,26,66,0	1,0,76,0	1,9,73,0	0,0,1	0,2,0	26,1,0	
122	10	0	0,0,119,1	0,9,121,1	1,0,76,0	1,9,73,0	0,0,1	9,1,1	26,1,0	
123	26	0	0,0,119,1	0,9,121,1	0,10,123,1	1,9,73,0	0,0,1	9,1,1	10,2,1	
125	0	0	0,0,119,1	0,9,121,1	0,10,124,1	0,26,124,1	26,3,1	9,1,1	10,2,1	

## 4. Output Analysis

Tick 10) pid 0, vpn 0

- 1. since TLB is empty, need to get data from memory
- 2. searching for IPT in memory, all IPT slots are invalid  
⇒ returned phyPage value is 0 (page in)
- 3. corresponding TLBentry for vpn 0 in TLB is 0 (FIFO started)

Tick 13) pid 0, vpn 9

- 1. since there is no vpn 9 in TLB, need to get data from memory
- 2. searching for IPT in memory, IPT[1] is invalid  
⇒ returned phyPage value is 1 (page in)
- 3. corresponding TLBentry for vpn 9 in TLB is 1 (FIFO)

Tick 15) pid 0, vpn 26

- 1. since there is no vpn 26 in TLB, need to get data from memory
- 2. searching for IPT in memory, IPT[2] is invalid  
⇒ returned phyPage value is 2 (page in)
- 3. corresponding TLBentry for vpn 26 in TLB is 2 (FIFO)

Tick 20) pid 0, vpn 1

- 1. since there is no vpn 1 in TLB, need to get data from memory
- 2. searching for IPT in memory, IPT[3] is invalid  
⇒ returned phyPage value is 3 (page in)
- 3. corresponding TLBentry for vpn 1 in TLB is 0 (FIFO)

Tick 26) pid 0, vpn 0

- 1. since there is no vpn 0 in TLB, need to get data from memory
- 2. searching for IPT in memory, there is vpn 0 in IPT[0]
- 3. corresponding TLBentry for vpn 0 in TLB is 1 (FIFO)

Tick 28) pid 0, vpn 10

- 1. since there is no vpn 10 in TLB, need to get data from memory
- 2. searching for IPT in memory, all slots are valid
  - ⇒ delete the page according to least recently used algorithm ; the lastUsed in vpn 26 is minimum value(17), and vpn 26 is in IPT[2] (page out)
  - ⇒ returned phyPage value is 2 (page in)
- 3. corresponding TLBentry for vpn 10 in TLB is 2 (FIFO)

Tick 41) pid 0, vpn 9

- 1. since there is no vpn 9 in TLB, need to get data from memory
- 2. searching for IPT in memory, there is vpn 9 in IPT[1]
- 3. corresponding TLBentry for vpn 9 in TLB is 0 (FIFO)

Tick 42) pid 0, vpn 26

- 1. since there is no vpn 26 in TLB, need to get data from memory
- 2. searching for IPT in memory, all slots are valid
  - ⇒ choose the page according to least recently used algorithm ; the lastUsed in vpn 1 is minimum value(22), and vpn 1 is in IPT[3]
  - ⇒ returned phyPage value is 3 (page in)
- 3. corresponding TLBentry for vpn 26 in TLB is 1 (FIFO)

Tick 47) pid 0, vpn 0

- 1. since there is no vpn 0 in TLB, need to get data from memory
- 2. searching for IPT in memory, there is vpn 0 in IPT[0]
- 3. corresponding TLBentry for vpn 0 in TLB is 2 (FIFO)

Tick 59) pid 1, vpn 0

- 1. since the process ID has changed(context switch), all TLB valid bit sets to 0
  - ⇒ need to get data from memory
- 2. searching for IPT in memory, all slots are valid
  - ⇒ choose the page according to least recently used algorithm ; the lastUsed in vpn 10 is minimum value(28), and vpn 10 is in IPT[2]
  - ⇒ returned phyPage value is 2 (page in)
- 3. corresponding TLBentry for vpn 10 in TLB is 0 (FIFO started)

Tick 62) pid 1, vpn 9

- 1. since there is no vpn 9 in TLB, need to get data from memory
- 2. searching for IPT in memory, all slots are valid
  - ⇒ delete the page according to least recently used algorithm ; the lastUsed in vpn 26 is minimum value(44), and vpn 26 is in IPT[3] (page out)
  - ⇒ returned phyPage value is 3 (page in)
- 3. corresponding TLBentry for vpn 9 in TLB is 1 (FIFO)

Tick 64) pid 1, vpn 26

- 1. since there is no vpn 26 in TLB, need to get data from memory
- 2. searching for IPT in memory, all slots are valid
  - ⇒ choose the page according to least recently used algorithm ; the lastUsed in vpn 9(\*pid 0) is minimum value(46), and vpn 9 is in IPT[1]
  - ⇒ returned phyPage value is 1 (page in)
- 3. corresponding TLBentry for vpn 26 in TLB is 2 (FIFO)

Tick 69) pid 1, vpn 1

- 1. since there is no vpn 1 in TLB, need to get data from memory
- 2. searching for IPT in memory, all slots are valid
  - ⇒ choose the page according to least recently used algorithm ; the lastUsed in vpn 0(\*pid 0) is minimum value(49), and vpn 0 is in IPT[0]
  - ⇒ returned phyPage value is 0 (page in)
- 3. corresponding TLBentry for vpn 1 in TLB is 0 (FIFO)

Tick 74) pid 1, vpn 0

- 1. since there is no vpn 0 in TLB, need to get data from memory
- 2. searching for IPT in memory, there is vpn 0 in IPT[2]
- 3. corresponding TLBentry for vpn 0 in TLB is 1 (FIFO)

Tick 117) pid 0, vpn 0

- 1. since the process ID has changed(context switch), all TLB valid bit sets to 0
  - ⇒ need to get data from memory
- 2. since process 1 has deleted, all IPT valid bit sets to 0
  - ⇒ returned phyPage value is 0 (page in)
- 3. corresponding TLBentry for vpn 0 in TLB is 0 (FIFO started)

Tick 120) pid 0, vpn 9

- 1. since there is no vpn 9 in TLB, need to get data from memory
- 2. searching for IPT in memory, IPT[1] is invalid  
⇒ returned phyPage value is 1 (page in)
- 3. corresponding TLBentry for vpn 9 in TLB is 1 (FIFO)

Tick 122) pid 0, vpn 10

- 1. since there is no vpn 10 in TLB, need to get data from memory
- 2. searching for IPT in memory, IPT[2] is invalid  
⇒ returned phyPage value is 2 (page in)
- 3. corresponding TLBentry for vpn 10 in TLB is 2 (FIFO)

Tick 123) pid 0, vpn 26

- 1. since there is no vpn 26 in TLB, need to get data from memory
- 2. searching for IPT in memory, IPT[3] is invalid  
⇒ returned phyPage value is 3 (page in)
- 3. corresponding TLBentry for vpn 26 in TLB is 0 (FIFO)

Tick 125) pid 0, vpn 0

- 1. since there is no vpn 0 in TLB, need to get data from memory
- 2. searching for IPT in memory, there is vpn 0 in IPT[0]
- 3. corresponding TLBentry for vpn 0 in TLB is 1 (FIFO)

(End)