CS 180 *Quiz #6* Name GOH WEI ZHE

DigiPen Institute of Technology November 30, 2020

Complete all questions. Total marks is 32.

1. Which of the following are 2 disadvantages of contiguous memory allocation? (Circle 2 answers only) [2 marks]

   (a) External fragmentation

   (b) Internal fragmentation

   (c) Can allocate memory only when a free chunk of contiguous memory block of large enough size exists

   (d) Memory access slower than paging

   (e) A more complex MMU design compared to paging

2. Write down 2 disadvantages of paging scheme for memory management. (2 marks, 1 for each correct answer)

   slow and expensive. memory

   Internal fragmentation, need extra space for paging table (PTE)

   Answer:

3. Which of the following statement regarding to virtual memory management is false? [1 mark]

   (a) Frame is the fixed-size basic unit of physical memory allocation and page is the fixed-size logical unit of process address space

   (b) A page fits in a frame but any page can be put in any frame

   (c) Reference bit is used to indicate if a page has been accessed (recently) and dirty bit indicates if a page has been modified (relative to disk)

   (d) Reference and dirty bits influence decision-making in page replacement policies

   (e) none of the above

4. Compare and contrast two approaches of paging and segmentation to memory management. Which of the following fails to identify the strengths and weaknesses of each? [1 mark]

   (a) Paging separates physical organization from logical address space and uses fixed-size units called pages, which are stored in page frames. It requires page table to keep track of (possibly many) pages.

   (b) Segmentation preserves user's structural view of logical address space and uses variable-size units called segments.

   (c) Segmentation can go anywhere in memory and requires segment table to keep track of the (few) segments. It needs base and limit register for each segment.

(d) These two techniques cannot be combined. Can combine.

(e) none of the above

5. To avoid excessively large page tables, many virtual memory systems have multi-level page tables. For this problem, we would like to figure out the details of a 2-level paging system with the following characteristics:

   - Both virtual and real (physical) memory addresses are 32 bits each
   - Pages are 4096 bytes each → 4KB - offset 12 bit
   - Page table entries are 4 bytes each, containing the physical page frame number, plus additional control bits including the valid bit.
   - The top-level page table consists of a single 4096-byte page and has as many page table entries as will fit. Page directory

   The page table structure and address translation can be diagrammed as follows (PPN, VPN1 and VPN2 are respectively frame number, Page Directory Number and page number in our lecture notes):

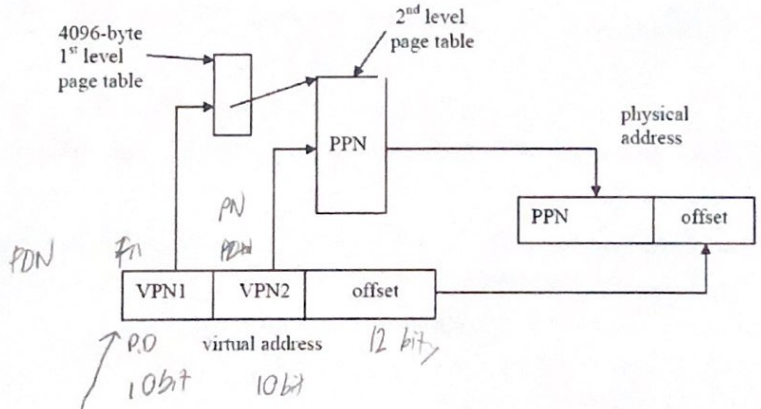


Figure 1: 2-Levl Paging

(a) Number of entries in the first level page table ? [1 mark]

Answer: 1024 4KB / 4B 10^2 = 1024

(b) Number of bits in the VPN1 part of the virtual address? [1 mark]

Answer: 10

(c) Number of bits in the offset field of each address [1 mark]

Answer: 12

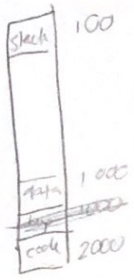
(d) Number of bits in the VPN2 part of the virtual address [1 mark]

Answer: 10

(e) Number of page table entries in each 2nd level page table [1 mark]

Answer: 1024 $2^{10} = 1024$

(f) Suppose that a process' virtual address space is structured, with program code at the "bottom" of the address space, program data (the heap) immediately following the code, and the stack at the "top". Suppose that the program code occupies 2,000 pages, the data occupies 1,000 pages, and the stack occupies 100 pages in the address space - the remainder of the address space is unused. How many frames will be occupied by the page table(s) for this process? [2 marks]



2000 / 1024 = 2 level-2 page table (code)
1000 / 1024 = 1 level-2 page table (Data)
100 / 1024 = 1 level-2 page table (Stack)

Answer: 250 Page Directory → 1 level-2 page table (Page Directory)

6. Consider the following diagram showing the logical and physical memory views of process P1. The diagram shows the starting address of each of the memory regions in the logical memory view – unallocated regions, text section, data section etc. For this question, we assume that the size of physical memory address is 4 bytes, which is the same as the size of logical address, and the page size is 4KB. 2nd level page tables are not shown in physical view.

In the diagram, the page numbers are reported **as if** the logical addresses are read in 1-level paging. That is those page numbers are taken from the first 20 bits of the logical address.

(a) How many pages are within data section? Please answer in base 10. [1 mark]

Answer: ~~1024~~ ~~22~~ 64 $0x101 = 257_{10}$

(b) How many pages are within text section? Please answer in base 10. [1 mark]

Answer: 18 $0x12 = 18_{10}$

(c) Assuming it's 2-level paging, write down all the page directory numbers and page numbers that are present respectively. Please answer in hexadecimal. [4 marks]

Answer: logical view (1) PDN = 0x0, PN = 0x10-0x21, 0x200-0x3FF;
(2) PDN = 0x1, 0x200-0x2FF;
(3) PDN = 0x4, 0x0-0x100, but present only
(1) PDN = 0x0, 0x200;
(2) PDN = 0x1, 0x200 - 0x2FF
(3) PDN = 0x4, 0x0

Data = 0x01000000 = 0x1000 = 0001 0000 0000 0000 → PDN = 4

text = 0x00010000 = 0x10 = 00 | 00 0001 0000 → PDN = 0

Stack + Heap = 0x00200000 = 0x200 = 0010 0000 0000 → PDN = 0

C Library = 0x00600000 = 0x600 = 0110 0000 0000 → PDN = 1

(d) Assuming 2-level paging, how many page tables should OS maintain for P1 (text section, stack, heap, C library, data), excluding page directory? [3 marks]

Answer: P.D.N = 0 (text segment, stack + heap)
P.D.N = 1 (C library)
P.D.N = 4 (data)
Total : 3 page table

(e) Assuming 2-level paging, if the frame numbers for P1 are shown in the diagram what can you observe for the page replacement from the diagram? [1 mark]

Answer: Text segment is not running, its not present in physical memory. (sleep), page out.

P1 is not in the running state as it is not present. (text section)

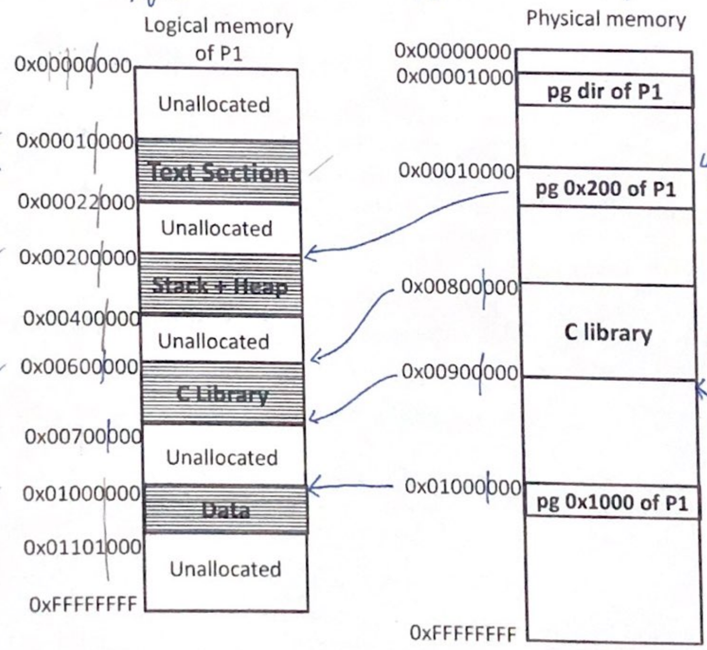


Figure 2: Logical and Physical Memory View of P1

7. In pure on-demand paging, a page replacement policy is used to manage system resources. Suppose that a newly-created process has 3 page frames allocated to it, and then generates the page references indicated below.

(a) How many page faults would occur with FIFO page replacement? [3 marks]

0x1000, PDN = 4: 0001 0000 0000 0000 | ...

A B C B A D A B C D A B A C B D

(3)

Answer: 12 /

F#	A	B	C	B	A	D	A	B	C	D	A	B	A	C	B	D
1	A	A	A	A	A	D	D	D	C	C	C	B	B	B	B	B
2		B	B	B	B	B	A	A	A	D	D	D	D	C	C	C
3			C	C	C	C	C	B	B	B	A	A	A	A	A	D
PF	Y	Y	Y	N	N	Y	Y	Y	Y	Y	Y	Y	N	Y	N	Y

(b) How many page faults would occur with LRU page replacement? [3 marks]

A B C B A D A B C D A B A C B D

(3)

Answer: 10 /

F#	A	B	C	B	A	D	A	B	C	D	A	B	A	C	B	D
1	A(0)	A(0)	A(0)	A(0)	A(4)	A(4)	A(6)	A(6)	A(6)	D(9)	D(9)	D(9)	D(9)	C(13)	C(13)	C(13)
2		B(1)	B(1)	B(1)	B(1)	B(1)	B(7)	B(7)	B(7)	B(7)	A(10)	A(10)	A(12)	A(12)	A(12)	D(15)
3			C(2)	C(2)	C(2)	D(5)	D(5)	D(5)	C(8)	C(8)	C(8)	B(11)	B(11)	B(11)	B(14)	B(14)
PF	Y	Y	Y	N	N	Y	N	N	Y	Y	Y	Y	N	Y	N	Y

(c) How many page faults would occur with Second chance (Clock) page replacement?[3 marks]

A B C B A D A B C D A B A C B D

Answer: 12 /

* scn pointer

F#	A	B	C	B	A	D	A	B	C	D	A	B	A	C	B	D
1	A+	A+	*A+	*A+	*A+	*D+	D+	*D+	C+	C+	*C+	B+	B+	*B+	B+	*B+
2		B+	B+	B+	B+	*B-	A+	A+	*A-	D+	D+	*D-	*D-	C+	C+	C+
3			C+	C+	C+	C-	*C-	B+	B-	*B-	A+	A-	A-	*A-	*A-	D+
PF	Y	Y	Y	N	N	Y	Y	Y	Y	Y	Y	Y	N	Y	N	Y

X X X

A	C	B	D
B+	B+	B+	*B-
*D-	C+	C+	C-
A-	*A+	*A+	D+
Y	Y	N	Y

3 cases

// int a [10000];

for (i=0 ; i<10000 ; i++) 4KB page

a[i] = a[i] * 2 ;

sleeping barber
chopstick

Scheduling 1 Qn -

Synchronisation 1 Qn - semaphore + test&set

MMU 1 Qn - today Quiz Q6

Page replacement (1)

algorithm

thread (1)

MCQ 40 points