# CS-341

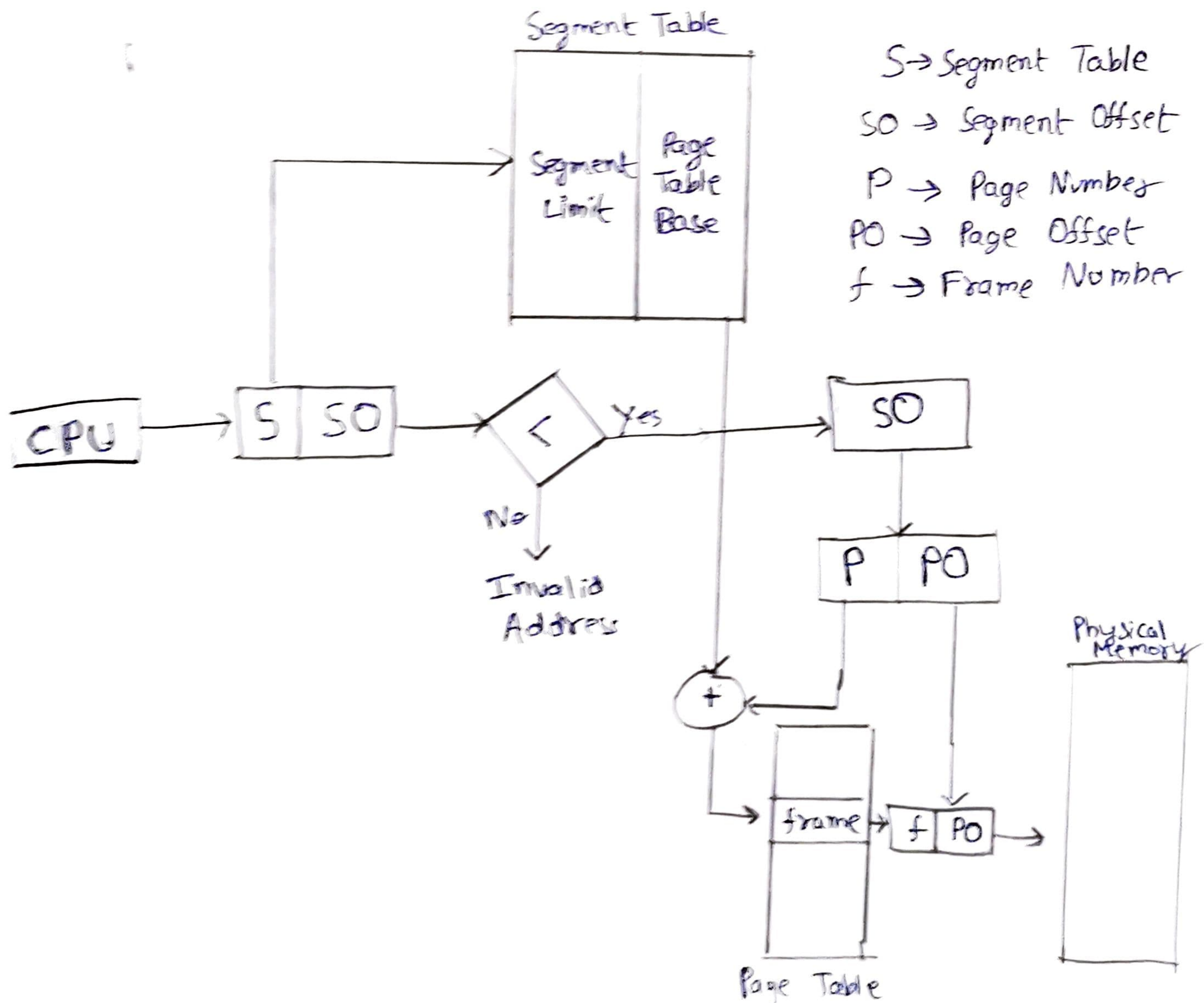
Assignment -1

Name : P.V.Sriram

Roll No. : 1801CS37

Q1) A certain computer system has the segmented paging architecture for virtual memory. The memory is byte addressable. Both virtual and physical address contain $2^{16}$ bytes each. The virtual address space is divided into 8 non-overlapping equal size segments. The memory management unit has a hardware segment table, each entry of which contains the physical address of page table. Page table entry size is 2bytes. What is the minimum page size in bytes so that the page table for a segment requires at most one page to store it? What is division of virtual address

ans)



Given,

Virtual Address Space = $2^{16}$ bytes

Physical Address Space = $2^{16}$ bytes

No. of segments of each process = 8

Size of one page table entry = 2 bytes

Now, we know that virtual address space is to be divided into 8 segments.

In the worst case, lets assume whole address space is occupied.

There fore, one segment = $\frac{2^{16}}{8} = 2^{13} = 8192$ bytes

Now this memory is to be allocated into pages in such a way that the page table is accomodated into single page.

Assume page size = $x$ bytes

$\Rightarrow$ No. of required pages = $\frac{8192}{x}$ entries. Now these entries are to be accomodated into single page.

$$\Rightarrow \quad \frac{8192}{x} \times \underset{\substack{\uparrow \\ \text{each entry size}}}{2} < x \Rightarrow 16384 < x^2 \Rightarrow \boxed{x > 128 \text{ bytes}}$$

Segment size = 8 bytes = $2^3$ = 3 bits required

Page size = 128 = $2^7$ = 7 bits required (Offset)

No. of pages = $2^{13}/2^7 = 2^6$ = 6 bits required

| Segment No. | Page No. | Page Offset |
|---|---|---|
| ← 3 bits → | ← 6 bits → | ← 7 bits → |

∴ Total 16 bits required ⇒ 3+6+7

---

Q2) Consider a machine with 64 MB physical and a 32 bit virtual address space. If the page size is 4 KB, what is the approximate size of the page table.

Ans) To find the size of the page table, we need number of entries and size of each entry / frame.

Entries number

Given, 32 bit virtual address space.

Now, we need this space to be segregated into pages.

$$\text{No. of pages} = \frac{\cancel{64\,MB}\ 2^{32} \text{ (Total space)}}{4\,KB \text{ (Page size)}} = \frac{2^{32}}{2^{12}} = 2^{20}$$

∴ For the page table we need $2^{20}$ entries

Entry size

Now, given that physical address is 64MB in size.

To segregate this into frames → $64 = 2^{26}/2^{12} = 2^{14}$ page frames.

i.e) 14 bits are required to address this space. But since its Byte addressable, 2 bytes (16 bits) are required.

$\Rightarrow$ Size of page Table = (Size of entry) × (No. of entries)

$$= 2^{20} \times 2 \text{ Bytes}$$

$$= \boxed{2\,MB}$$

---

Q3) In a virtual memory system, size of virtual address is 32 bit, size of physical address is 30 bit, page size is 4KB and size of each page table entry is 32 bit. The main memory is byte addressable. Which one of the following is the maximum number of bits that can be used for storing protection and other information in each page table entry?

Ans) Given,

Virtual memory is 32 bit address space

Physical memory is 30 bit address space

Page size = 4KB = $2^{12}$ bytes

No. of frames = Physical Memory Size / Page size

$= 2^{30} / 2^{12} = 2^{18}$

i.e) We need 18 bits to represent this space

Although, we have a 32 bit virtual address space

∴ we have 32 - 18 = 14 bits to represent other information.

$\boxed{14 \text{ bits}}$

Q4) Consider a system using a multi level paging scheme. The page size is 1 MB. The memory is byte addressable and the virtual address is 64 bits long. The page table entry size is 4 bytes.

a) How many levels of page table will be required?

b) Give the divided physical address and virtual address.

ans) We have,

Page table entry size = 4 bytes = 32 bits

$\therefore$ Number of bits in frame number = 32 bits

We get no. of bits in frame number = 32 bits

No. of frames in main memory = $2^{32}$ frames.

Size of main memory

= Total no. of frames x frame size

= $2^{32}$ x 1 MB = $2^{52}$ B

$\therefore$ No. of bits in physical address = 52 bits

We have page size = 1 MB = $2^{20}$ B

$\therefore$ No. of bits in page offset = 20 bits

No. of bits in virtual address = 64 bits

$\therefore$ Process size $= 2^{64}$ bits

No. of pages the process is divided

$= $ process size / page size

$= 2^{64}B / 1MB = 2^{64}/2^{20} = 2^{44}$ pages

Inner page table size

$=$ No. of entries $\times$ Page table entry size

$= 2^{44} \times 4$ bytes $= 2^{46}$ byte

Size of inner page table > Size of a single frame.

Hence, further paging is required

Now,

No. of page the inner page table is divided

$=$ inner page table size / Page size

$= 2^{46}B / 2^{20}B = 2^{26}$ pages

No. of page table entries in one page of inner page Table

$= 2^{20}/2^{2} = 2^{18}$ entries.

Now, for outer page table -1

Size $=$ No. of entries $\times$ page table entry size

= No. of pages of inner page table × entry size.

$= 2^{26} \times 4$ bytes

= 256 MB.

As 256 MB > 1 MB hence further paging is required.

Number of pages of outer page table -1

= 256 MB / 1 MB = 256 pages.

Also, no. of entries in one page of outer page table -1

$$= \frac{\text{Page Size}}{\text{Entry Size}} = \frac{1\,MB}{4\,B} = 2^{18}$$

∴ 18 bits are required to represent one entry in one page of outer page table-1.

As mentioned before, we divide outer page table 1 into pages using outer page table 2.

Size of outer page table 2 = No. of entries × Entry size

$= 256 \times 4B = 1\,KB$

Since, the size outer page table -2 is less than the frame size, it can be stored in a single frame.

Conclusion:

The given system has 3 levels of paging, with one inner table and 2 outer tables.

Since, the outer table has $2^8 = 256$ entries, 8 bits are required to represent one entry.

$$\text{Physical address} = \underbrace{32 \text{ bits}}_{\text{Frame entry Space}} + \underbrace{20 \text{ bits}}_{\text{Frame entry size}}$$

$$\text{Logical address} = \underbrace{8}_{\text{outer page table 2}} + \overbrace{18}^{\text{outer page table 1}} + \underbrace{18}_{\text{inner page table}} + \overbrace{20}^{\text{main memory}}$$

Q5) Given, a segmentation scheme with addressing consisting of two parts.

i.e) Segment Number/ Base, & Segment Offset.

Physical Address = Base + Offset

Segment Table:

| Segment No. | Base | Length |
|---|---|---|
| 0 | 1219 | 700 |
| 1 | 2300 | 14 |
| 2 | 90 | 100 |
| 3 | 1327 | 580 |
| 4 | 1952 | 96 |

Note: Offset must lie in the limits of length.

ie) For Segment 0 (Length 700) valid offsets are 0 to 699

Now, option A

$(0, 430) \rightarrow$ Segment 0 + Offset 430

430 is a valid offset ($< 700$)

$= 1219 + 430 \Rightarrow \boxed{1649}$

Now, option B

$(1, 11) \rightarrow$ Segment 1 + Offset 11

11 is valid offset ($< 14$)

$= 2300 + 11 = \boxed{2311}$

Now, option C

$(2, 100) \rightarrow$ Segment 2 + Offset 100

100 is Invalid Offset ($= 100$)

$\therefore$ $\boxed{\text{Invalid}}$ (Trap addressing error)

Now, option D

(3, 425) → Segment 3 + 425 Offset

425 is valid offset (<580)

$1327 + 425 = \boxed{1752}$

Now, option E

(4, 95) → Segment 4 + 95 Offsets

95 is valid Offset (<96)

$1952 + 95 = \boxed{2047}$