27-03-21

CS341 Assignment-1

M. Maheeth Reddy
1801CS31

Ans 1 It is given that:

→ Virtual address space contains 2^{16} bytes.

→ It is divided into 8 equal size segments

→ Physical address space contains 2^{16} bytes.

→ Page table contains 2^{byte} entries.

We need to find:

(*) Minimum page size so that page table for a segment requires atmost one page to store it.

(**) Division of virtual address.

$$\text{Size of each segment} = \frac{\text{Size of virtual address space}}{8}$$

$$= \frac{2^{16}}{8} = 2^{13} \text{ bytes} = 8\text{KB}$$

$$\therefore \boxed{\text{Size of each segment} = 2^{13} \text{ bytes} = 8\text{KB}}$$

It has been mentioned that page table has to be stored into a single page.

Hence, $\boxed{\text{Size of page table} \leq \text{Page Size}}$

Assume page size = N bytes

Now, no. of pages in each segment = $\frac{\text{Segment Size}}{\text{Page Size}}$

$$= \frac{8K \text{ bytes}}{N \text{ bytes}} = \frac{8K}{N}$$

∴ Size of each page table = Entry size × no. of entries

= 2 bytes × {no. of pages in each segment}

⇒ Size of each page table = $2 \times \frac{8K}{N}$ bytes = $\frac{16K}{N}$ bytes

As per condition, Size of page table ≤ Page Size

$$\Rightarrow \frac{16K}{N} \leq N$$

$$\Rightarrow N^2 \geq 16K \Rightarrow N^2 \geq 2^4 \times 2^{10}$$

$$\Rightarrow N \geq 2^2 \times 2^5 = 2^7$$

$$\Rightarrow \boxed{N \geq 2^7} \Rightarrow \boxed{N \geq 128} \text{ bytes}$$

Minimum page size possible is 128 bytes

Division of virtual address:

Virtual address consists of Segment No., Page No., Page Offset.

* Since there are 8 segments, we need 3-bits to identify a particular segment

* No. of pages in a segment = $\frac{8K}{128} = \frac{2^3 \times 2^{10}}{2^7} = 2^6$

Hence, 6-bits are required to identify a page in segment

* Size of a page is 128 bytes. To represent offset value we need 7-bits ($128 = 2^7$)

So

Size of virtual address	=	Size of Segment No.	+	Size of Page No.	+	Size of Page offset
16-bits	=	3-bits	+	6-bits	+	7-bits

Ans 2: Given that,

Physical memory = 64 MB

Virtual address size = 32-bit

Page Size = 4KB = 2^{12} B

We have to find approx. size of page table.

Physical address space = 64 MB = $2^6 \times 2^{20}$ B = 2^{26} B

∴ No. of bits to represent physical address = 26-bits

$$\text{No. of pages} = \frac{\text{Virtual address space}}{\text{Page Size}} = \frac{2^{32}}{2^{12}} = 2^{20}$$

$$\text{No. of frames} = \frac{\text{Physical memory}}{\text{Page Size}} = \frac{2^{26}\text{B}}{2^{12}\text{B}} = 2^{14}$$

Page table only consists of frame number. So, page table size depends on frame number. Since, no. of frames is 2^{14}, Page table has 14 bits along with 2 bits to represent valid and invalid.

Hence, page table has 16 bits

Page Table Size = Entry Size × No. of pages

= 16-bits × 2^{20}

= 2^{20} × 2 bytes

Page Table Size = 2 MB

Ans 3 Given,

Size of virtual address = 32-bit

Size of physical address = 30-bit

Page size = 4 KB = 2^{12} B

Size of page table entry = 32-bit

$$\text{No. of frames} = \frac{\text{Physical memory}}{\text{Frame Size}} = \frac{2^{(\text{Size of physical address})}}{2^{12}}$$

$$= \frac{2^{30}}{2^{12}} = 2^{18}$$

Hence, 18-bits is size of frame address

Page table entries contain Bits for frame addresses and other info mentioned in question.

Page table entry is 32-bit ⇒ **For other information, (32-18) = 14 bits are required**

Ans 4: Given,

Page Size = 1MB

Size of virtual address = 64 bits

Size of page table entry = 4 bytes

Length of Frame Number } = Size of page table entry = 4 bytes

= 32-bits

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

⇒ Size of main memory = No. of frames × Frame Size

= 2^{32} × 1MB

(Page Size × frame size are same)

⇒ **Size of main memory = $2^{32} \times 2^{20} = 2^{52}$ bytes**

∴ Physical address has 52-bits

Since, page size = 1MB = 2^{20}B,

no. of bits in page offset = 20 bits

no. of bits in virtual address = 64 bits

∴ Physical memory or Process Size } = 2^{64} bytes

Hence, no. of pages in this process = $\frac{\text{process size}}{\text{page size}} = \frac{2^{64}}{2^{20}} = 2^{44}$

Since, this is a multilevel paging scheme, there is an inner page table which is kept track of through further outer page tables.

We have to have outer page tables such that the size of the outer page table is less than the frame size.

So, size of inner page table = Entry size × No. of entries

$= 4 \text{ bytes} \times 2^{44}$

$= 2^{46} \text{ bytes}$

Since, inner page can't be stored in one frame ($2^{46}B > 1MB$) it is divided into pages as mentioned previously.

No. of pages in inner page table

$= \frac{\text{Inner page table size}}{\text{Page Size}} = \frac{2^{46}B}{1MB} = \frac{2^{46}}{2^{20}} = \underline{\underline{2^{26}}}$

∴ No. of bits required to search an entry in one page of inner page table = 18-bits

[∵ Each page of inner page table has $\frac{\text{Page Size}}{\text{Entry Size}} = \frac{2^{20}}{2^2} = 2^{18}$ entries]

Now, outer page table 1 size

= No. of entries × Entry size

= No. of pages in inner page table × Entry size

= $2^{26} \times 4$ bytes = 2^{28} bytes = 256 MB

Clearly, the outer page table 1 can't be stored in a single frame (256 MB > 1 MB). It has to be divided into pages

No. of pages in outer table-1 = $\frac{\text{Size of Outer page table-1}}{\text{Page Size}}$

= $\frac{256\text{MB}}{1\text{MB}}$ = 256 pages.

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

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

∴ 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 × 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 & 2 outer tables.

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

Division of physical address (52-bits) = 32-bits + 20-bits

32-bits → no. of frames

20-bits → no. of frame size

Division of logical address, 64-bits

= 8-bits (entries in outer page table 2)
+ 18-bits (entry in ~~inner~~ outer page table 1)
+ 18-bits (entry in inner page table)
+ 20-bits (addressing main memory)

Ans 5 The logical address consists of 2 parts:

① Segment Number

② Segment Offset

The value of segment offset ranges from 0(zero) to (length of segment − 1). If the offset value is not in this range, trap addressing error is produced.

In A. 0, 430 :

Segment number is 0 ⇒ Length is 700

430 lies between 0 and (700−1)

Hence, trap address error won't occur.

Physical address = Base address + Segment Offset

= 1219 + 430

⇒ Physical Address = 1649

In B. 1, 11 :

Segment number is 1 ⇒ Length is 14

11 lies between 0 and (14−1) ⇒ So, no trap error.

Physical address = Base address (2300) + Segment Offset (11) = 2311

In C: 2,100

Segment number is 2 ⇒ Length is 100

100 (offset given) DOESN'T lie between 0 and (100-1)

So, trap addressing error occurs

In D: 3,425

Segment number is 3 ⇒ Length is 580

425 lies between 0 and (580-1)

So, trap is not produced.

Physical address = Base Address + Segment offset

= 1327 + 425

Physical Address = 1752

In E: 4, 95

Segment number is 4 ⇒ Length is 96

95 Lies between 0 and (96-1) ⇒ So, trap is not produced

Physical address = Base address + Segment offset

= 1952 + 95 = 2047

⇒ Physical Address = 2047