Virtual Memory and Paging (1) (Practice)

ICS332

Operating Systems

Henri Casanova (henric@hawaii.edu)

(q1) Address bits

- A computer has 32 GiB of RAM with a page size of 16KiB; Processes can have up to 4 GiB address spaces
  - How many bits are used for physical addresses?
  - How many bits are used for logical addresses?
  - How many bits are used for logical page numbers?

(q1) Answers

- A computer has 32 GiB of RAM with a page size of 16KiB; Processes can have up to 4 GiB address spaces
  - How many bits are used for physical addresses

    Physical RAM: 32 GiB = 2^{35} bytes

    → 35-bit physical addresses
  - How many bits are used for logical addresses

    Logical address space: 4 GiB = 2^{32} bytes

    → 32-bit physical addresses
  - How many bits are used for logical page numbers?

    Page size = 2^{14} bytes

    Number of pages in logical address space: $2^{32}/2^{14} = 2^{18}$

    → 18-bit logical page numbers
    (and 14-bit offsets)

(q2) Page Size

- Logical addresses are 44-bit, and a process can have up to 2^{27} pages
- What is the page size?

(q2) Answer

- Logical addresses are 44-bit, and a process can have up to 2^{27} pages
- What is the page size?

We simply need to divide the total number of bytes in the address space by the number of pages:

The page size is $2^{44} / 2^{27} = 2^{17}$ bytes

(q3) Address bits

- On my computer the page size is 16 KiB, and a process' address space can be up to is 4GiB
- How many bits are used for the page number in a logical address?

(q3) Answer

- On my computer the page size is 16 KiB, and a process' address space can be up to is 4GiB
- How many bits are used for the page number in a logical address?

The address space contains 2^{32} bytes

A page is 2^{14} bytes

Therefore, my address space has $2^{32}/2^{14} = 2^{18}$ pages

Therefore, we need 18 bits for the page number in a logical address (and we have 14 bits in the offset)

(q4) Page table

- A computer has 32-bit physical addresses. The logical page number in a logical address is 14-bit. A process can have up to a 2GiB address space
- Let's consider a process with currently a 1GiB address space (i.e., it could get up to another 1GiB during execution).
- What is the page size?
- How many entries in the process' page table currently point to pages that are part of the address space?

(q4) Answer

- A computer has 32-bit physical addresses. The logical page number in a logical address is 14-bit. A process can have up to a 2GiB address space
- Let's consider a process with currently a 1GiB address space (i.e., it can get up to another 1GiB during execution).
- What is the page size?

Bytes in 2GiB (the max address space): 2^{31}
Therefore: 31-bit logical addresses
Therefore: 31 - 14 = 17-bit offsets
Therefore: 2^{17} bytes in a page
Therefore: 128KiB pages

- How many entries in the process' page table currently point to pages that are part of the address space?

The process has a 1GiB = 2^{30}-byte address space
Number of pages in the address space: $2^{30}/2^{17} = 2^{13}$
Therefore: there are 2^{13} entries in the page table, each pointing to one page
(makes sense: 2^{14} would be the number of pages for a 2GiB address space)

(q5) RAM size

- Logical addresses are 40-bit, and a process can use at most 1/4 of the physical RAM
  - How big is the RAM?
  - A process has at most 2^{22} pages on this system. How many bits are used for the “offset” part of logical addresses?

(q5) Answer

- Logical addresses are 40-bit, and a process can use at most 1/4 of the physical RAM
  - How big is the RAM?

With 40-bit logical addresses, an address space is at most 2^{40} bytes

So the RAM is 4 times as big: 2^{42} bytes, which is 4TiB

  - A process has at most 2^{22} pages on this system. How many bits are used for the "offset" part of logical addresses?

Since we have 2^{22} pages, 22 bits are used for the page number

Therefore 40 - 22 = 18 bits are used for the offset

(q6) Entry size

- Consider a system that uses 14-bit logical addresses, and has 256-byte pages
- Consider a process with an address space as large as possible
- For that process, we know that the page table occupies exactly half a page
- What is the page table entry size, in bytes?

(q6) Answer

- Consider a system that uses 14-bit logical addresses, and has 256-byte pages
- Consider a process with an address space as large as possible
- For that process, we know that the page table occupies exactly half a page
- What is the page table entry size, in bytes?

- The process has $2^{14} / 2^{8} = 2^{6}$ pages
- Half a page is 2^{7} bytes
- The number of entry in the page table is 2^{6}
- Therefore, each entry is $2^{7}/2^{6} = 2$ bytes

(q7) Page Table

- Consider a system with 4-byte pages. A process has the following entries in its page table:

logical	physical
0	4
1	5
2	30

- What is the physical address of the byte with logical address 2?
- What is the physical address of the byte with logical address 9?

(q7) Answer 1

- Consider a system with 4-byte pages. A process has the following entries in its page table:

logical	physical
0	4
1	5
2	30

- What is the physical address of the byte with logical address 2?
- The byte with logical address 2 is the 3rd byte in page 0 (because that page contains the bytes at addresses 0, 1, 2, and 3)
- Page 0, according to the page table is in physical frame 4
- The first byte of physical frame 0 is at physical address 4 × 0 = 0 (the first byte in physical RAM)
- The first byte of physical frame 1 is at physical address 4 × 1 = 4 (the fifth byte in physical RAM)
- …
- The first byte of physical frame 4 is at physical address 4 × 4 = 16
- The 3rd byte of physical frame is thus at address 16 + 2
- Therefore, the byte at logical address 2 is at physical address 18

(q8) Answer 2

- Consider a system with 4-byte pages. A process has the following entries in its page table:

logical	physical
0	4
1	5
2	30

- What is the physical address of the byte with logical address 9?
- The byte with logical address 9 is in page 9 / 4 = 2 (integer division)
- Its offset on that page is 9 % 4 = 1
- Page 2 is in frame 30
- Therefore, the byte at logical address 9 is at physical address 30 x 4 + 1 = 121

(q8) Address Translation

- Assume a page size of 1000 bytes
- If logical address 2040 successfully translates to physical address 54040, then what is the corresponding page table entry in the page table?
- Could logical address 5052 ever translate to physical address 24123?

(q8) Answer

- Assume a page size of 1000 bytes
- If logical address 2040 successfully translates to physical address 54040, then what is the corresponding page table entry in the page table?
  - Entry 2:54 (page 2 is in frame 54)
- Could logical address 5052 ever translate to physical address 24123?
  - No (the offset would be the same, and it's 52 for the logical address and 123 for the physical address)