Virtual Memory

Review

One Process

Recall that after:

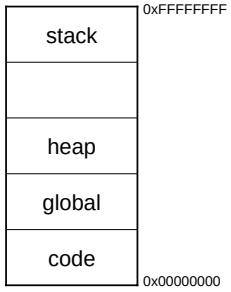
(1) compilation

(2) assembly

(3) linkage

we get an executable with absolute addresses.

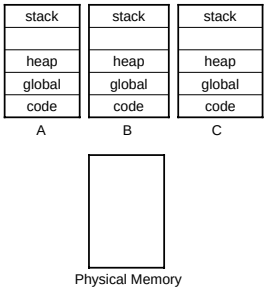
This is the address space of a single process.



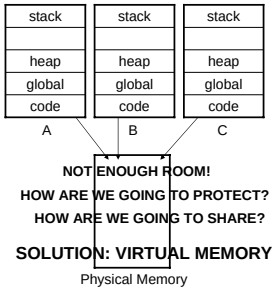
> One Process

What happens when we have more than one process running at the same time?

Each process needs its own address space, but we only have *one* physical memory.



> One Process



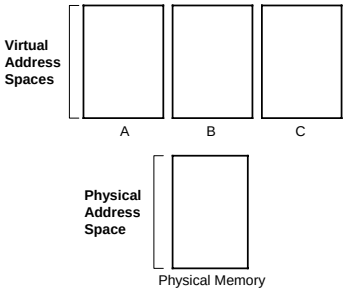
NOT ENOUGH ROOM!

HOW ARE WE GOING TO PROTECT?

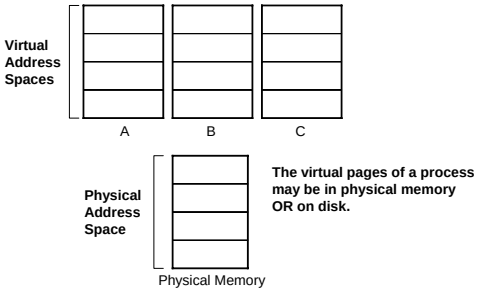
HOW ARE WE GOING TO SHARE?

SOLUTION: VIRTUAL MEMORY

> One Process

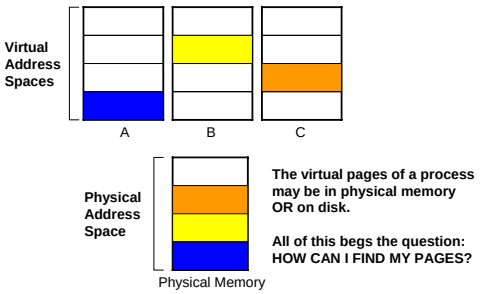


> One Process



The virtual pages of a process may be in physical memory OR on disk.

> One Process



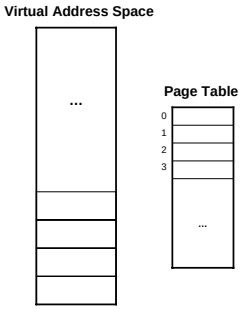
Page Table

The Question:

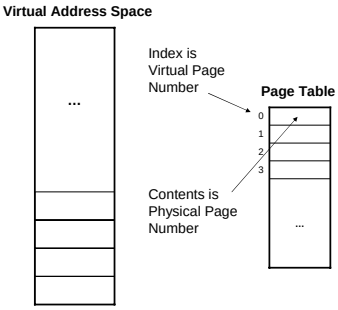
How can I find my pages?

The Answer:

Give each process a page table that is used to translate virtual addresses to physical addresses

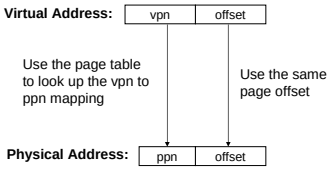


Page Table



Page Table

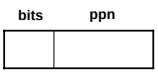
Use the page table to translate virtual addresses to physical addresses.



Page Table

What other information could we store in the page table?

- Valid bit (required)
- Physical page number (required)
- Dirty bit (required)
- Referenced bit
- Access rights



Why don't we store the virtual page number? What are the other bits used for? Why is the dirty bit required?

Page Table

Why don't we store the virtual page number?

The vpn is the index for the page table.

What are the other bits used for?

Valid → 1 if in memory, 0 if on disk

Dirty → has the page been written to?

Reference → used to approximate LRU

Access Rights → read only, etc.

Why is the dirty bit required?

Memory is always write back, so upon eviction we need to know whether to write the page back to disk.

Ex1

What does it mean when the valid bit is zero?

Ex1

What does it mean when the valid bit is zero?

The page is on disk.

Ex2

Virtual addresses: 40 bits
Physical addresses: 30 bits
Page size: 8 KB

How many bits are used for the page offset, virtual page number, and physical page number?

Ex2

Virtual addresses: 40 bits
Physical addresses: 30 bits
Page size: 8 KB

How many bits are used for the page offset, virtual page number, and physical page number?

offset bits: 13, vpn bits: 27, ppn bits: 17

Ex3

Virtual addresses: 40 bits
Physical addresses: 30 bits
Page size: 8 KB

How many page table entries are required for a single process?

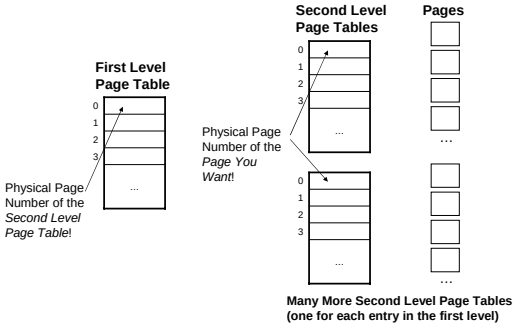
Ex3

Virtual addresses: 40 bits
Physical addresses: 30 bits
Page size: 8 KB

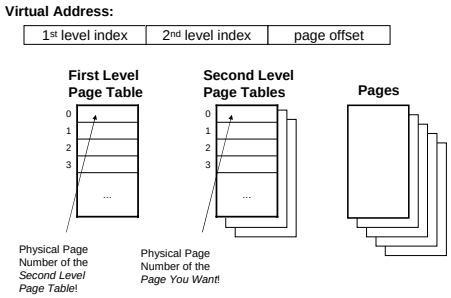
How many page table entries are required for a single process?

2^{27}. *Page Tables are too big, so use two level page tables!*

Two Level Page Table



Two Level Page Table



Ex4

The two level page table will have more entries than if we had only used a one level page table. How exactly are we saving space?

Ex4

The two level page table will have more entries than if we had only used a one level page table. How exactly are we saving space?

In the one level page table, the page table had to stay in memory all the time. In the two level page table, only the first level page table needs to stay in memory all the time… huge savings!

Ex5

Assuming our data is in main memory, how many times do we need to go to memory in order to do one memory access (assuming that all you have is a two level page table)?

Ex5

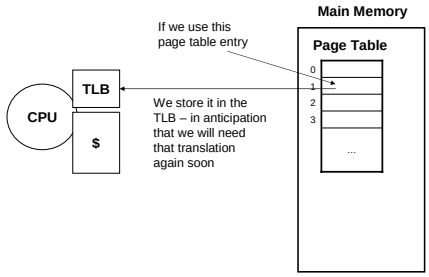
Assuming our data is in main memory, how many times do we need to go to memory in order to do one memory access (assuming that all you have is a two level page table)?

3. One for each level of the page table and one for the data itself.

This is too slow, so use a cache (TLB) to keep recently used page table entries.

TLB

The TLB is a cache for page table entries



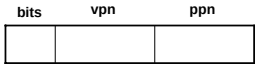
Ex6

What information should we store in the TLB?

Ex6

What information should we store in the TLB?

Everything we stored in the page table entry (valid, ppn, dirty, referenced, access rights) AND the vpn.



Ex7

The TLB is generally much smaller than the data cache, but the TLB will certainly have a higher hit rate than the data cache. Why?

Ex7

The TLB is generally much smaller than the data cache, but the TLB will certainly have a higher hit rate than the data cache. Why?

Memory accesses in the same page will use the same TLB entry.

Memory accesses in the same block will use the same cache entry.

Pages: 1KB-8KB. Blocks: 32B-64B.

Ex8

How (the heck) are page tables, the TLB, the data cache, main memory, and disk all related?

See lecture notes from 8/4.