Memory Management

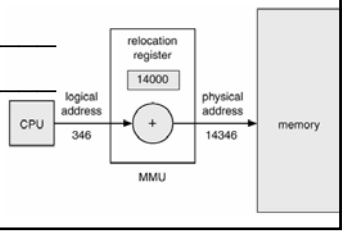
- Reading: OS Concepts pp.273-316
- Contents
  - Introduction
  - Contiguous
  - Paging
  - Segmentation

Background

- To be run a program must be _____
  _____
- Memory must be shared due to multiprocessing
- Problems
  - Address binding: _____
    _____
  - _____
    _____

Binding & Address Space

- Binding of instructions and data can happen at
  - Compile time: _____
  - Load time: _____
  - Execution time: _____
- _____
  _____
  - _____
  - _____
    _____

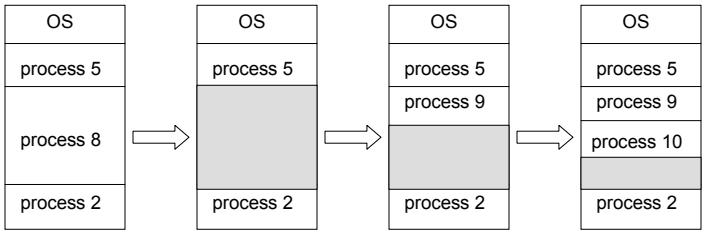


Dynamic Loading and Linking

- Dynamic loading
  - _____
  - _____
  - _____
- Dynamic Linking
  - _____
  - _____
  - _____
- Example: _____
- Problem: _____
  _____

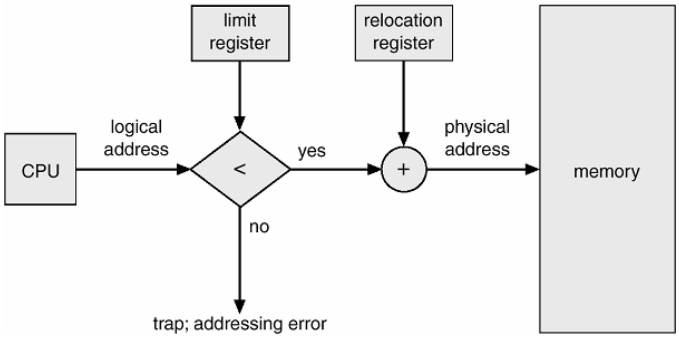
Contiguous Allocation

- Main memory split into the operating system and user processes
- User process partition may be given to one process OR multiple processes. If multiple processes are allowed then memory consists of:
  - _____
  - _____



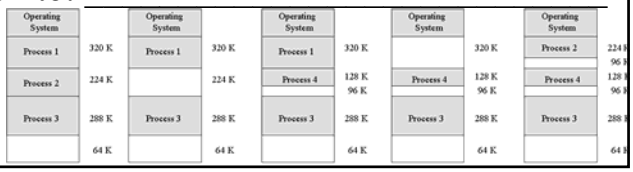
Base and Limit Registers

- Execution time binding is implemented using a relocation register scheme to protect processes from each other. For each process we have a:
  - Base register: _____
  - Limit register: _____



Fitting & Fragmentation

- How to satisfy a request for memory:
  - First fit: _____
  - Best fit: _____
  - Worst fit: _____
- Memory Fragmentation
  - External: _____
  - Internal: _____
  - Compaction to: _____

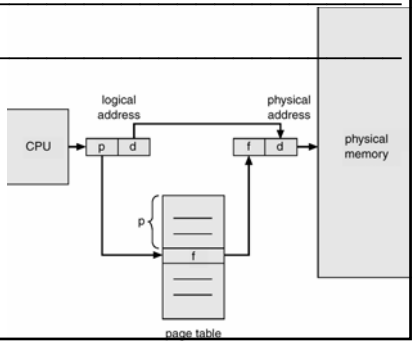


Paging

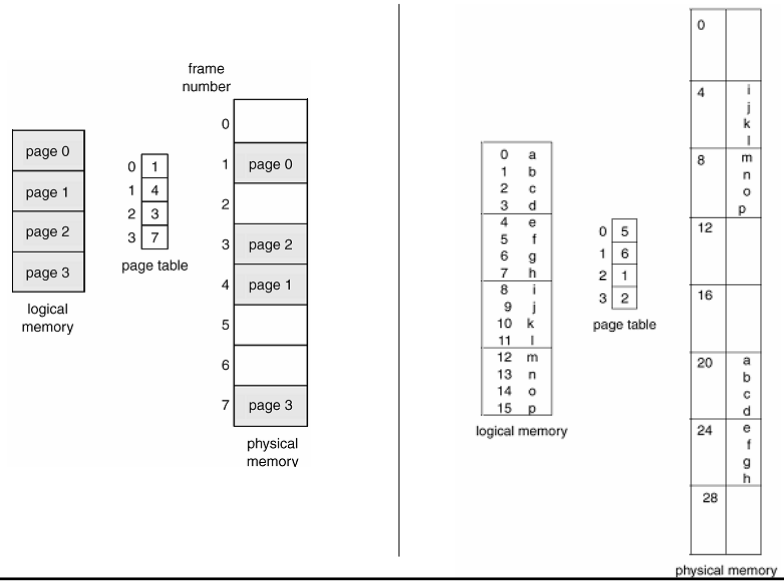
- Divide Physical Memory into **frames** (of fixed size)
- Divide Logical Memory into **blocks** (of fixed size)
- _____
- _____
- _____
- _____
- Address divided into
  - Page Number

    _____
  - Page Offset

    _____

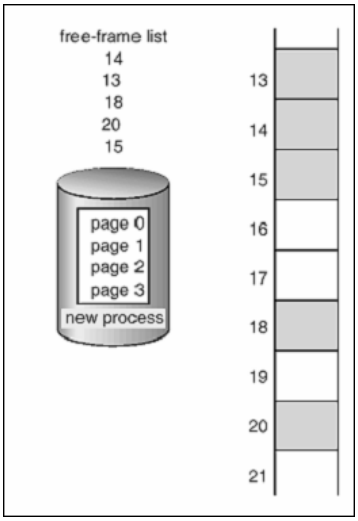


Paging Example



Paging Free frames

- Free frame list



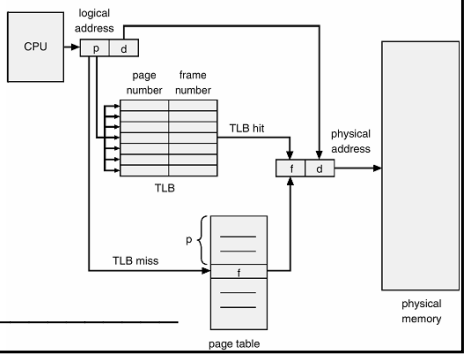
Paging Implementation

- Page table is kept is main memory
  - Base Register: _____
  - Limit Register: _____
- _____
  _____
- To overcome this use an associate memory
  - Known as a TLB (translation look-aside buffer)
  - _____
    _____
  - _____
    _____

Page #	Frame #

Paging Evaluation

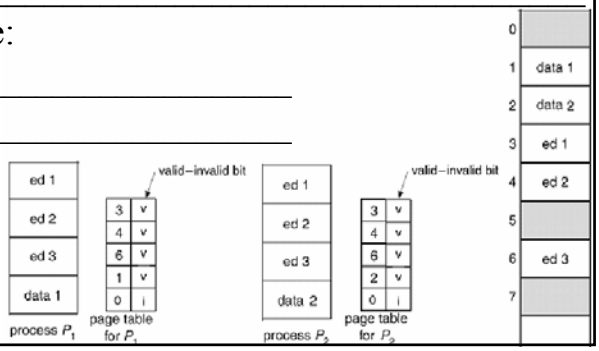
- EAT = $(m + \varepsilon)\ \alpha + ((n+1)*m + \varepsilon)(1 - \alpha)$
  - m: _____
  - ε: _____
  - α: _____
  - n: _____
- Example
  - m = 100ns
  - ε = 10ns
  - α = 0.98
  - n = 1
  - EAT = _____



Paging Memory Protection

- Memory protection using a valid-invalid bit for each frame for each process
  - _____
    _____
- _____
- _____

  They must be:
  - _____
  - _____



How to structure the page table

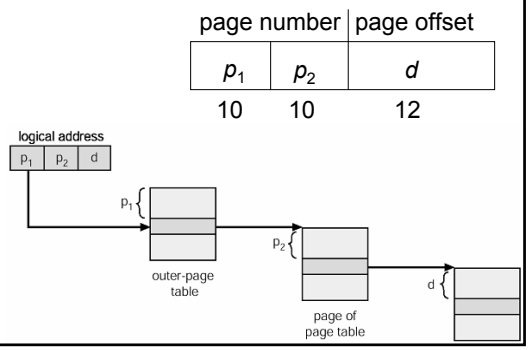
1. Hierarchical Paging

- Consider
  - 32 bit address space
  - 4K page size
  - Page offset = _____, Page number = _____
  - Page table size = _____
- Problem: _____

How to structure the page table

1. Hierarchical Paging (contd)

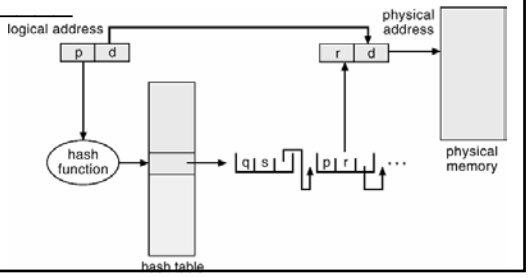
- To overcome this use multilevel/hierarchical paging
  - In this example: _____
    - _____
    - EAT = _____ which is a slowdown of _____



How to structure the page table

2. Hashed Page Tables

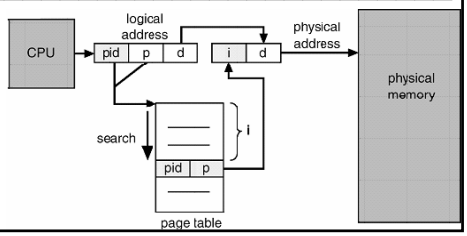
- Hash the logical page number into a hash table which contains
  - _____
    _____
  - _____
    _____
  - _____



How to structure the page table

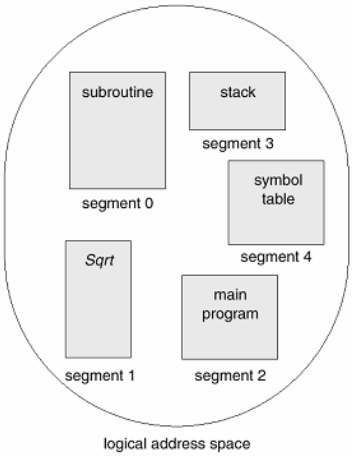
3. Inverted Page Tables

- There are significantly less frames than pages
- _____
  - _____
    - _____
    - _____
- Advantage: _____
- Disadvantage: _____
  - _____
  - _____



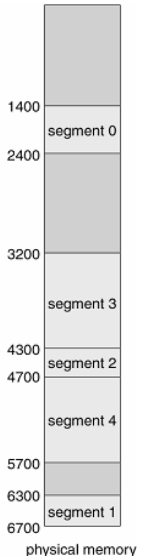
Segmentation

- _____



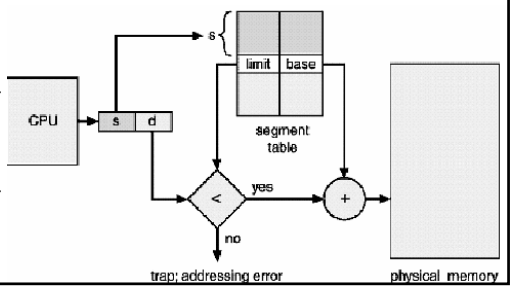
	limit	base
0	1000	1400
1	400	6300
2	400	4300
3	1100	3200
4	1000	4700

segment table



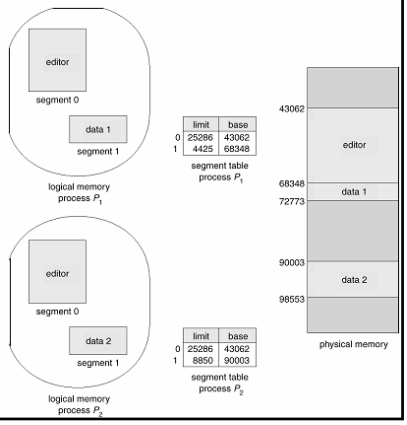
Implementing segments

- Logical address consists of segment number + offset
- Segment table maps logical segments to physical addresss. Each table entry has
  - Base: _____
  - Limit: _____
- Each process has
  - Base Register: _____
  - Limit Register: _____

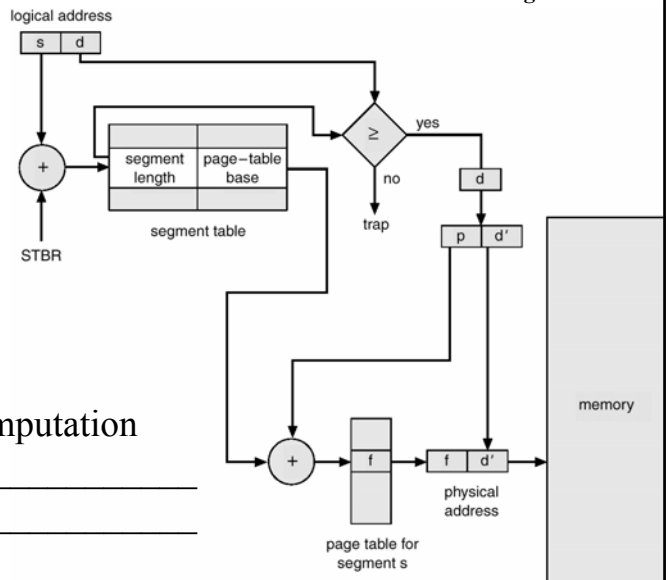


Segment protection, sharing & allocation

- Protection. For each segment maintain
  - _____
  - _____
- _____
  _____
- _____
  _____

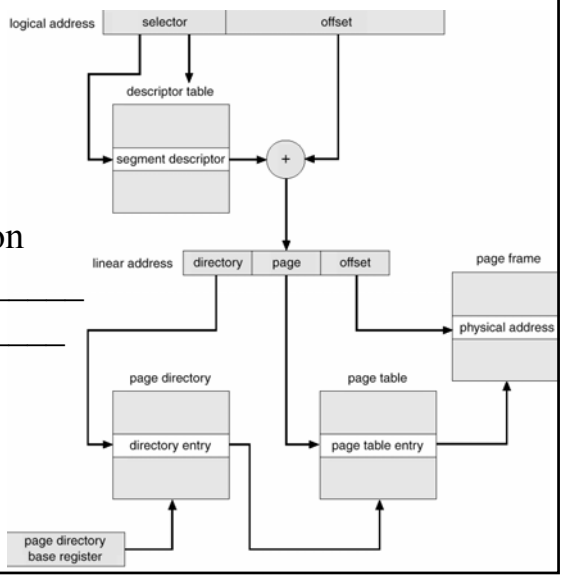


MULTICS



- EAT Computation

Intel 80386



- EAT Computation