Lesson 21, 22

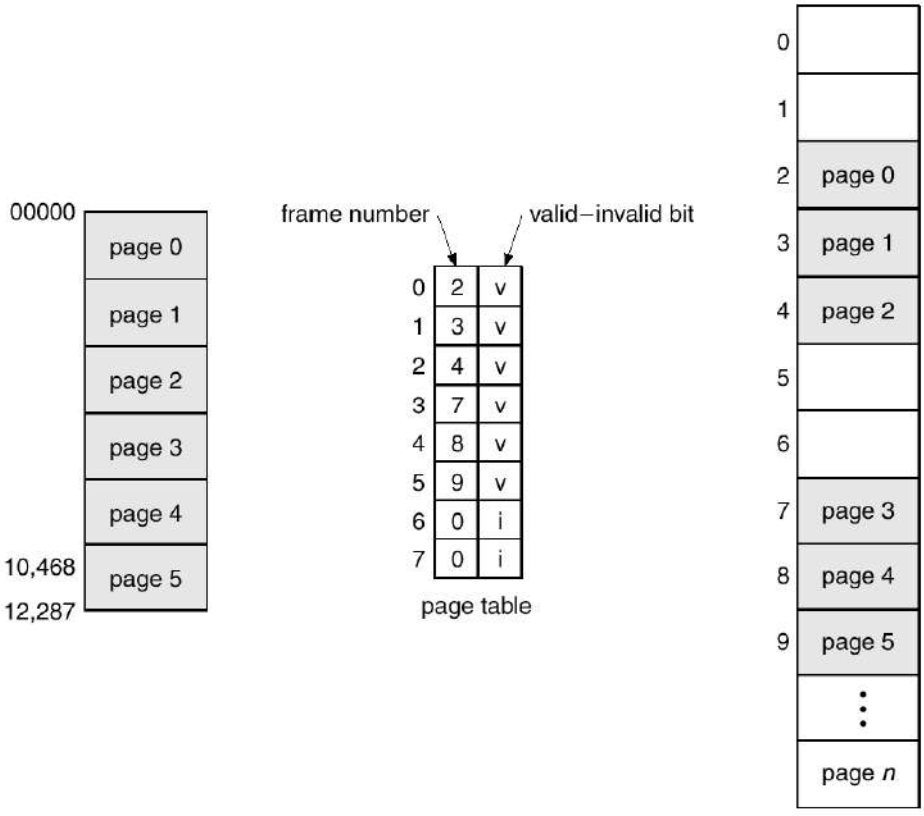
Objectives

- Memory Protection
- Page Table Structure
- Segmentation
- Segmentation Architecture
- Segmentation with Paging

Memory Protection

There must be some check if some invalid reference is invoked in page table.

- Memory protection implemented by associating protection bit with each frame.
- Valid-invalid bit attached to each entry in the page table:
  - "Valid" indicates that the associated page is in the process' logical address space, and is thus a legal page.
  - "Invalid" indicates that the page is not in the process' logical address space.



Page Table Structure

There are three possible architectures of page table.

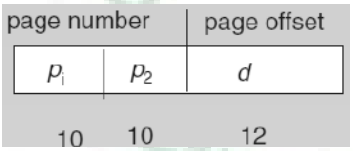
- Hierarchical Paging
- Hashed Page table
- Inverted Page table

Hierarchical Paging

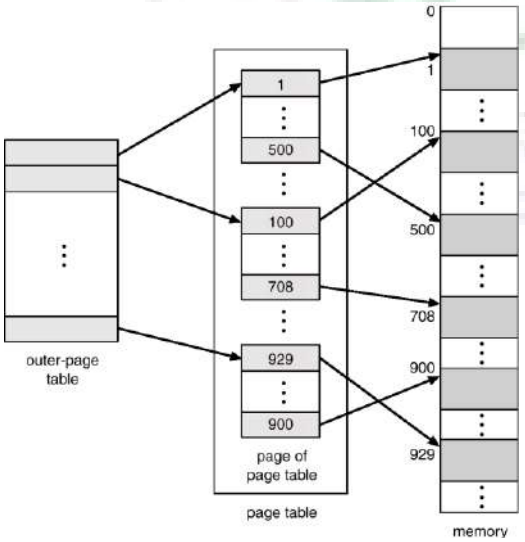
- Break up the logical address space into multiple page tables.
- A simple technique is a two-level page table.

Two level paging example

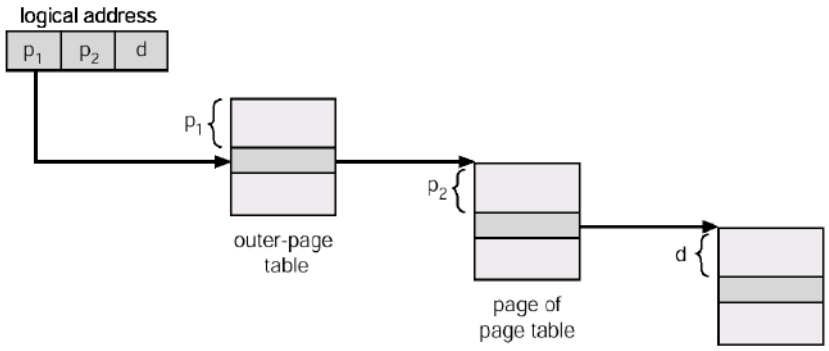
- A logical address (on 32-bit machine with 4K page size) is divided into:
  - A page number consisting of 20 bits.
  - A page offset consisting of 12 bits.
- Since the page table is paged, the page number is further divided into:
  - A 10-bit page number.
  - A 10-bit page offset.
- Thus, a logical address is as follows



Where Pi is an index into the outer page table, and p2 is the displacement within the page of the outer page table.

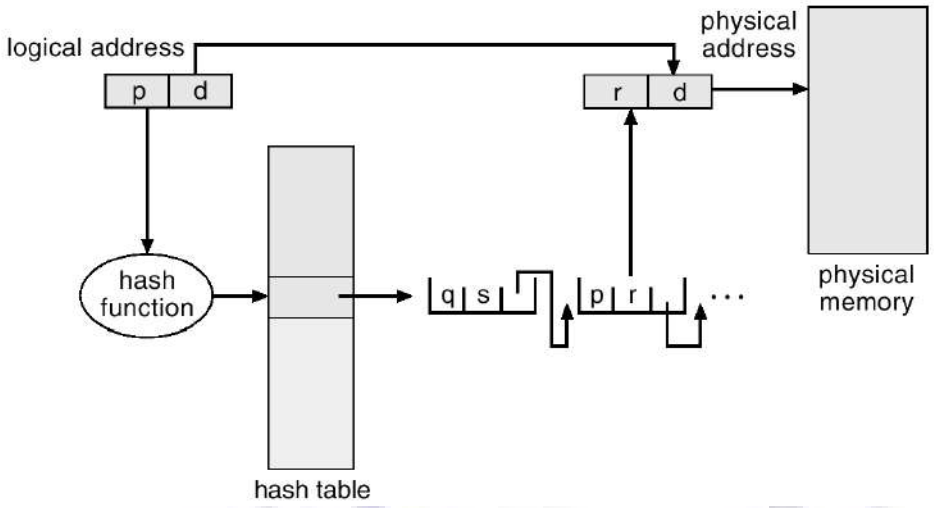


Address-translation scheme for a two-level 32-bit paging architecture.



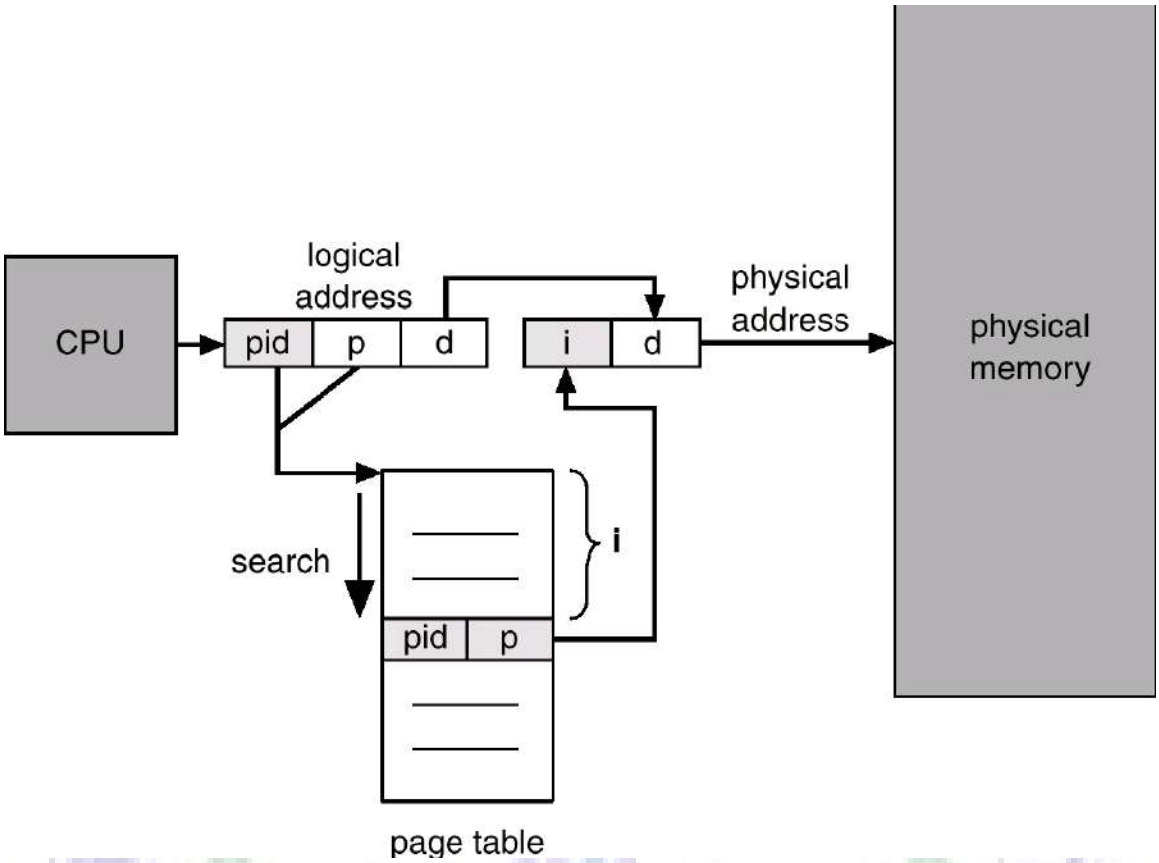
Hashed Page Table

- Common in address spaces > 32 bits.
- The virtual page number is hashed into a page table. This page table contains a chain of elements hashing to the same location.
- Virtual page numbers are compared in this chain searching for a match. If a match is found, the corresponding physical frame is extracted.



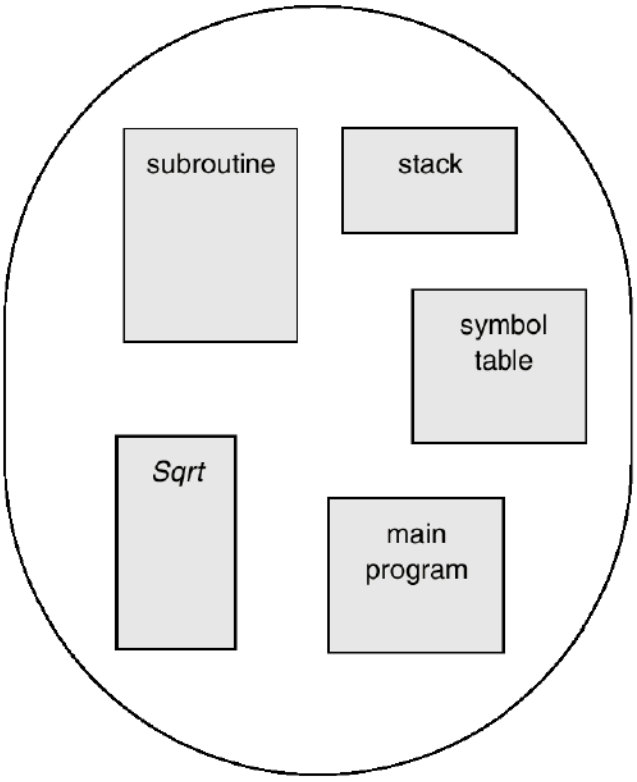
Inverted Page Tables

- One entry for each real page of memory.
- Entry consists of the virtual address of the page stored in that real memory location; with information about the process that owns that page.
- Decreases memory needed to store each page table, but increases time needed to search the table when a page reference occurs.
- Use hash table to limit the search to one — or at most a few — page-table entries.

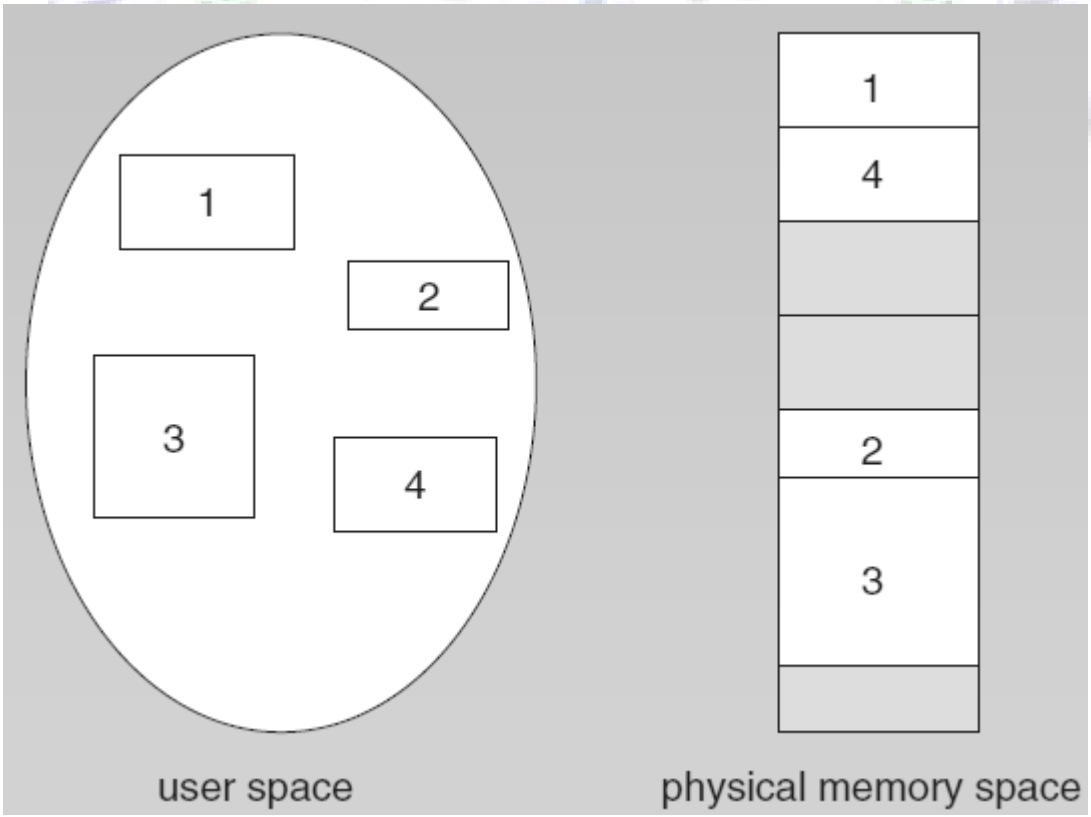


Segmentation

- Memory-management scheme that supports user view of memory.
- A program is a collection of segments. A segment is a logical unit such as:
  - main program,
  - procedure,
  - function,
  - method,
  - object,
  - local variables, global variables,
  - common block,
  - stack,
  - symbol table, arrays

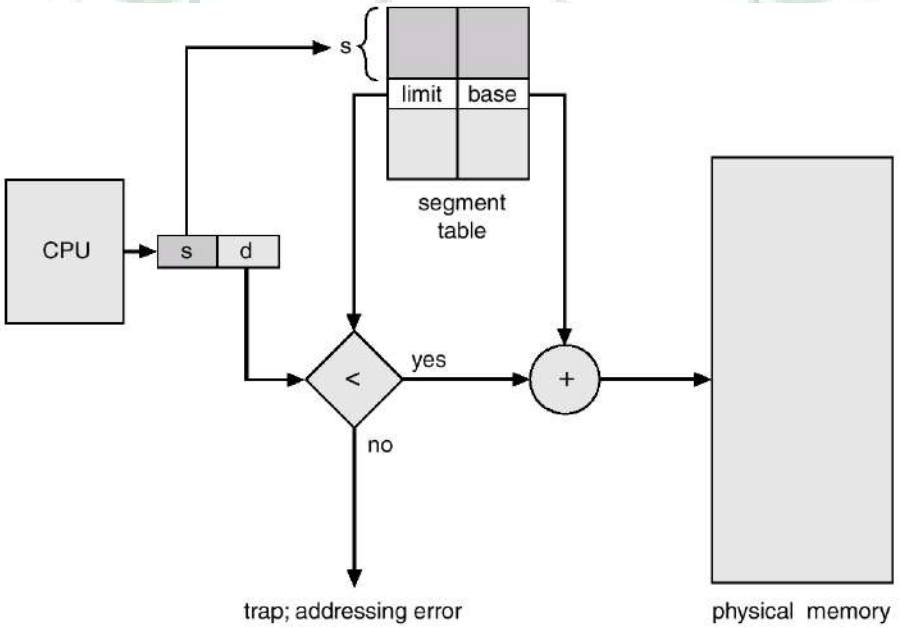


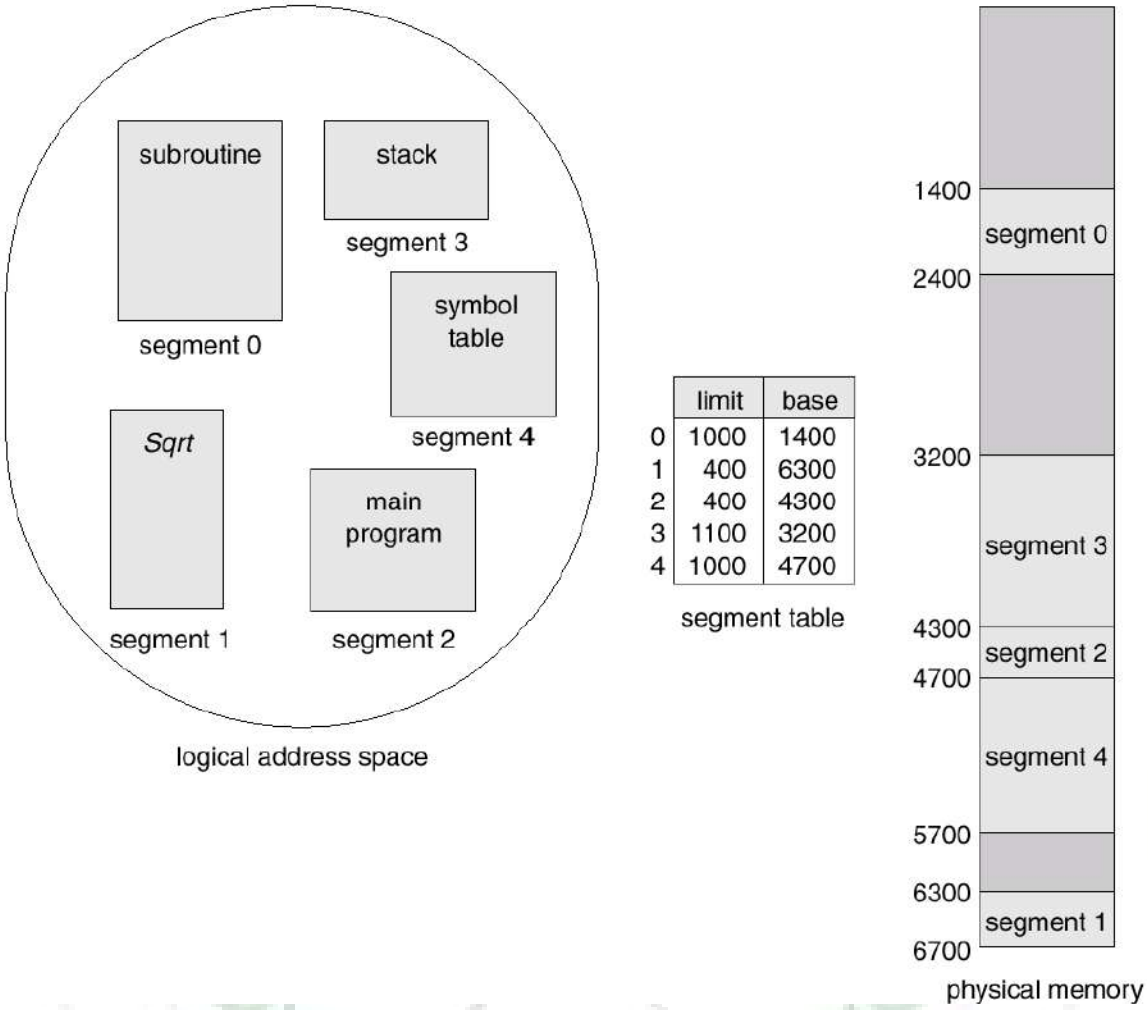
logical address space



Segmentation Architecture

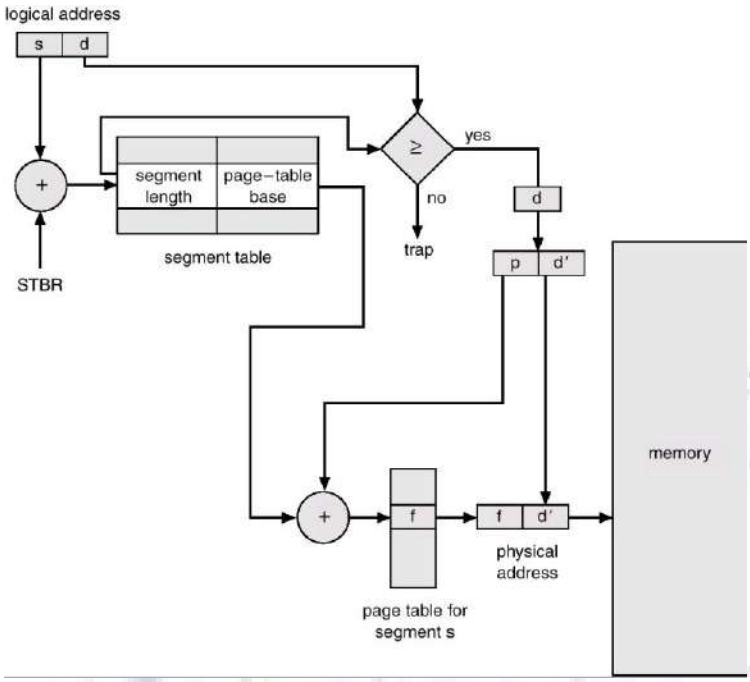
- Logical address consists of a two tuple:
  <segment-number, offset>,
- Segment table – maps two-dimensional physical addresses; each table entry has:
  - Base – contains the starting physical address where the segments reside in memory.
  - Limit – specifies the length of the segment.
- Segment-table base register (STBR) points to the segment table's location in memory.
- Segment-table length register (STLR) indicates number of segments used by a program; segment number s is legal if s < STLR.
- Protection. With each entry in segment table associate:
  - validation bit = 0 _ illegal segment
  - read/write/execute privileges
- Protection bits associated with segments; code sharing occurs at segment level.
- Since segments vary in length, memory allocation is a dynamic storage-allocation problem.
- A segmentation example is shown in the following diagram

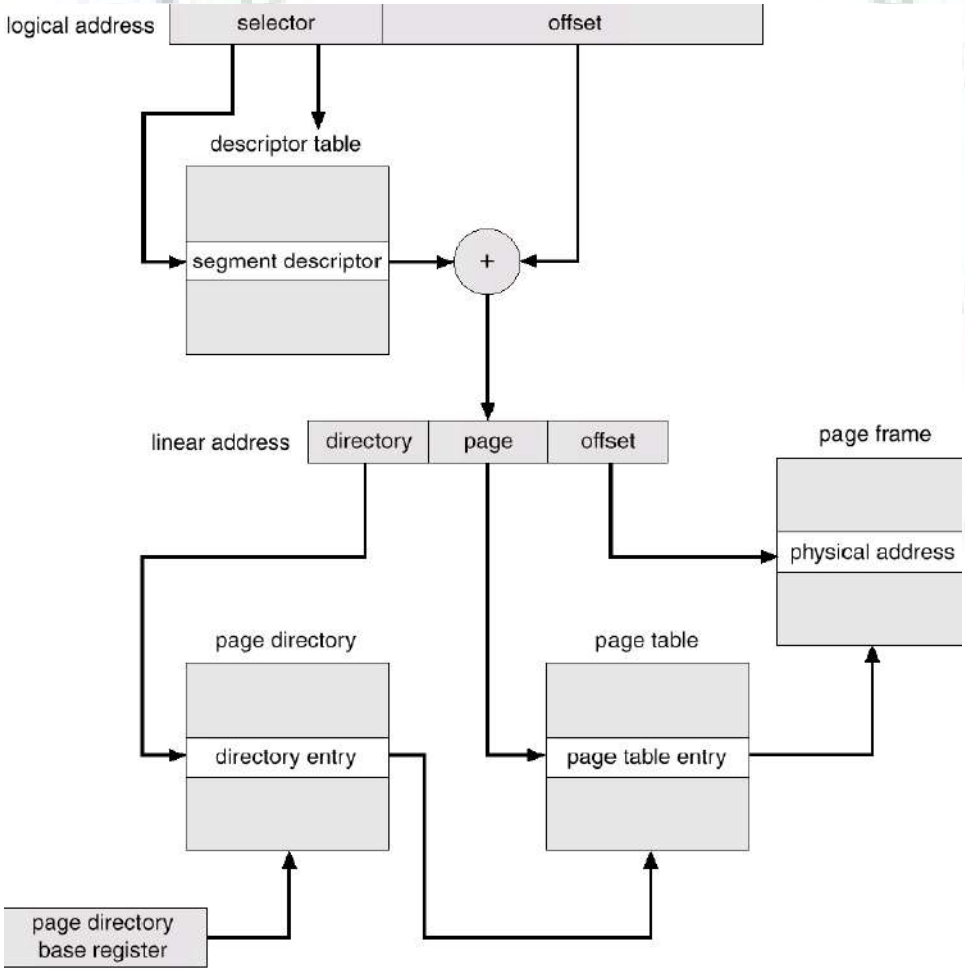




Segmentation with Paging

- The MULTICS system solved problems of external fragmentation and lengthy search times by paging the segments.
- Solution differs from pure segmentation in that the segment-table entry contains not the base address of the segment, but rather the base address of a page table for this segment.
- As shown in the following diagram, the Intel 386 uses segmentation with paging for memory management with a two-level paging scheme.


logical address
s
d
+
segment length
page-table base
segment table
STBR
≥
yes
no
trap
d
p
d'
memory
+
f
page table for segment s
f
d'
physical address


logical address
selector
offset
descriptor table
segment descriptor
+
linear address
directory
page
offset
page frame
physical address
page directory
directory entry
page table
page table entry
page directory base register