

## Segmentation

CS 537 – Introduction to Operating Systems

---

---

---

---

---

---

---

## Segmentation

- Segmentation is a technique for breaking memory up into logical pieces
- Each “piece” is a grouping of related information
  - data segments for each process
  - code segments for each process
  - data segments for the OS
  - etc.
- Like paging, use virtual addresses and use disk to make memory look bigger than it really is
- Segmentation can be implemented with or without paging

---

---

---

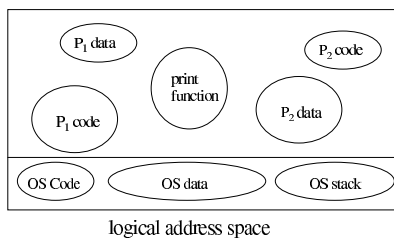
---

---

---

---

## Segmentation



---

---

---

---

---

---

---

## Addressing Segments

- Let's first assume no paging in the system
- User generates logical addresses
- These addresses consist of a segment number and an offset into the segment
- Use segment number to index into a table
- Table contains the physical address of the start of the segment
  - often called the base address
- Add the offset to the base and generate the physical address
  - before doing this, check the offset against a limit
  - the limit is the size of the segment

---

---

---

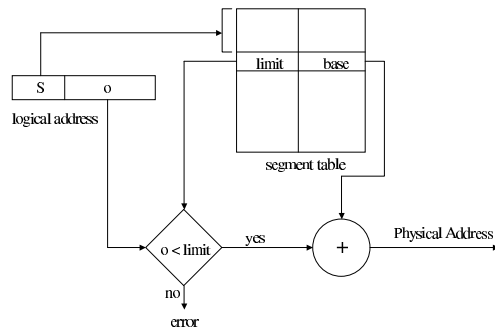
---

---

---

---

## Addressing Segments



---

---

---

---

---

---

---

## Segmentation Hardware

- Sounds very similar to paging
- Big difference – segments can be variable in size
- As with paging, to be effective hardware must be used to translate logical address
- Most systems provide segment registers
- If a reference isn't found in one of the segment registers
  - trap to operating system
  - OS does lookup in segment table and loads new segment descriptor into the register
  - return control to the user and resume
- Again, similar to paging

---

---

---

---

---

---

---

## Protection and Sharing

- Like page tables, each process usually gets its own segment table
- Unlike page tables, there usually exists a global segment table for everyone
  - this, however, is usually used by OS
- Access rights for segment are usually included in table entry
- Multiple processes can share a segment

---

---

---

---

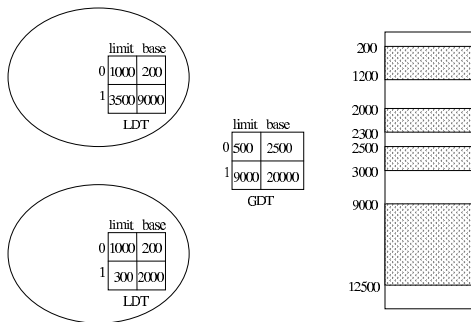
---

---

---

---

## Protection and Sharing




---

---

---

---

---

---

---

---

## Segmentation Issues

- Entire segment is either in memory or on disk
- Variable sized segments leads to external fragmentation in memory
- Must find a space big enough to place segment into
- May need to swap out some segments to bring a new segment in

---

---

---

---

---

---

---

---

## Segmentation with Paging

- Most architectures support segmentation and paging
- Basic idea,
  - segments exist in virtual address space
  - base address in segment descriptor table is a virtual address
  - use paging mechanism to translate this virtual address into a physical address
- Now an entire segment does not have to be in memory at one time
  - only the part of the segment that we need will be in memory

---

---

---

---

---

---

---

## Linear Address

- The base address gotten from the segment descriptor table is concatenated with the offset
- This new address is often referred to as a linear address
- This is the address that is translated by the paging hardware

---

---

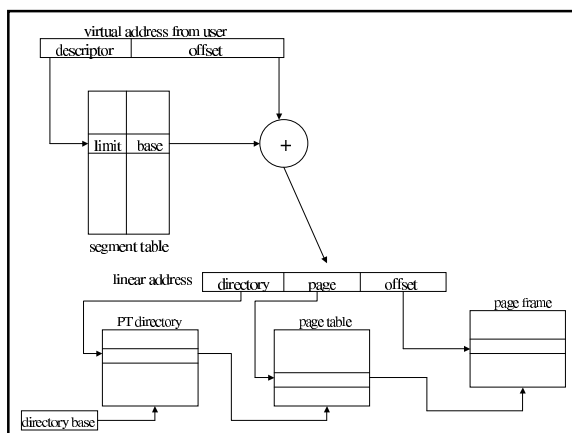
---

---

---

---

---



---

---

---

---

---

---

---