Overlays

- A programmer would divide their programs into pieces.
- The programmer determined which pieces never needed to be used at the same time.
- A portion of the program loaded from disk or stored to disk these pieces during execution.
- The programmer ensured that the maximum number of program pieces would fit into physical memory.
Virtual Memory

- Divides physical memory into blocks and allocates these blocks to different processes.
- Provides a mapping between blocks in physical memory and blocks on disk.
- Each process has its own virtual address space.
- Allows a process to execute with only portions of the process being in main memory.
- Also reduces program startup time.
- Provides protection to prevent processes from accessing blocks inappropriately.
Virtual Memory Terms

- *Page* is the name used for a block.
- *Page fault* is the name for a miss.
- *Virtual address* is the address produced by a CPU.
- *Physical address* is the address used to access main memory and typically cache as well.
- *Page table* is the data structure containing the mappings between virtual and physical addresses.
- *Translation Lookaside Buffer* is a cache that contains a portion of the page table.
Figure 7.20: In virtual memory, blocks of memory are mapped from one set of addresses, called virtual addresses, to another set called physical addresses.
Figure 7.21: Mapping from a virtual to a physical address.
### Table at Bottom of Page 541 Again

<table>
<thead>
<tr>
<th>Memory technology</th>
<th>Typical access time</th>
<th>$ per MByte in 1997</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRAM</td>
<td>5-25 ns</td>
<td>$100-$250</td>
</tr>
<tr>
<td>DRAM</td>
<td>60-120 ns</td>
<td>$5-$10</td>
</tr>
<tr>
<td>Magnetic Disk</td>
<td>10-20 million ns</td>
<td>$0.10-$0.20</td>
</tr>
</tbody>
</table>
Designing Virtual Memory Systems

- The page size should be large enough to avoid the high latency of disk access (32KB to 64KB are common for newly designed systems).
- Fully associative placement of pages is used to reduce the number of page faults.
- Page faults are handled by software (operating system) to use better algorithms for minimizing the number of page faults.
- A write-back policy is used (instead of write-through), so disk accesses can be reduced.
Page Tables

- Number of entries in page table is equal to the number of virtual pages.
- Number of virtual pages is the size of the virtual address space divided by the page size.
- Each process has its own page table.
- A page table register points to the beginning of the page table for the current process.
- A page table entry will typically contain:
  - physical page number
  - valid bit
  - dirty bit
  - use bit
  - protection field (e.g. read only)
  - disk address
Figure 7.22: The page table is indexed with the virtual page number to obtain the corresponding portion of the physical address.
Figure 7.23: The page table maps each page in virtual memory to either a page in physical memory or a page stored on disk, which is the next level in the hierarchy.
Translation Lookaside Buffers

- Most machines use special caches that contain a portion of the entries in a page table.
- Each entry in the TLB contains a tag (portion of the virtual page number) and most of the information in a page table entry.
- TLBs are typically cleared when a context switch is performed.
- TLBs are typically quite small to provide a very fast translation.
- There are typically separate TLBs for instructions and data.
Figure 7.24: The TLB acts as a cache on the page table for the entries that map to physical pages only.
Table 7.27: The possible combinations of events in the TLB, virtual memory system, and cache.

<table>
<thead>
<tr>
<th>Cache</th>
<th>TLB</th>
<th>Virtual memory</th>
<th>Possible? If so, under what circumstances?</th>
</tr>
</thead>
<tbody>
<tr>
<td>miss</td>
<td>hit</td>
<td>hit</td>
<td>Possible, although the page table is never really checked if TLB hits.</td>
</tr>
<tr>
<td>hit</td>
<td>miss</td>
<td>hit</td>
<td>TLB misses, but entry found in page table; data is found in cache.</td>
</tr>
<tr>
<td>miss</td>
<td>miss</td>
<td>hit</td>
<td>TLB misses, but entry found in page table; data misses in cache.</td>
</tr>
<tr>
<td>miss</td>
<td>miss</td>
<td>miss</td>
<td>TLB misses and is followed by a page fault; data must miss in cache.</td>
</tr>
<tr>
<td>miss</td>
<td>hit</td>
<td>miss</td>
<td>Impossible; cannot have a translation in TLB if page is not present in memory.</td>
</tr>
<tr>
<td>hit</td>
<td>hit</td>
<td>miss</td>
<td>Impossible; cannot have a translation in TLB if page is not present in memory.</td>
</tr>
<tr>
<td>hit</td>
<td>miss</td>
<td>miss</td>
<td>Impossible; data cannot be allowed in cache if the page is not present in memory.</td>
</tr>
</tbody>
</table>