

Table 8.1 Virtual Memory Terminology

| | |
|------------------------------|---|
| Virtual memory | A storage allocation scheme in which secondary memory can be addressed as though it were part of main memory. The addresses a program may use to reference memory are distinguished from the addresses the memory system uses to identify physical storage sites, and program-generated addresses are translated automatically to the corresponding machine addresses. The size of virtual storage is limited by the addressing scheme of the computer system and by the amount of secondary memory available and not by the actual number of main storage locations. |
| Virtual address | The address assigned to a location in virtual memory to allow that location to be accessed as though it were part of main memory. |
| Virtual address space | The virtual storage assigned to a process. |
| Address space | The range of memory addresses available to a process. |
| Real address | The address of a storage location in main memory. |

Table 8.2 Characteristics of Paging and Segmentation

| Simple Paging | Virtual Memory Paging | Simple Segmentation | Virtual Memory Segmentation |
|---|---|---|---|
| Main memory partitioned into small fixed-size chunks called frames | Main memory partitioned into small fixed-size chunks called frames | Main memory not partitioned | Main memory not partitioned |
| Program broken into pages by the compiler or memory management system | Program broken into pages by the compiler or memory management system | Program segments specified by the programmer to the compiler (i.e., the decision is made by the programmer) | Program segments specified by the programmer to the compiler (i.e., the decision is made by the programmer) |
| Internal fragmentation within frames | Internal fragmentation within frames | No internal fragmentation | No internal fragmentation |
| No external fragmentation | No external fragmentation | External fragmentation | External fragmentation |
| Operating system must maintain a page table for each process showing which frame each page occupies | Operating system must maintain a page table for each process showing which frame each page occupies | Operating system must maintain a segment table for each process showing the load address and length of each segment | Operating system must maintain a segment table for each process showing the load address and length of each segment |
| Operating system must maintain a free frame list | Operating system must maintain a free frame list | Operating system must maintain a list of free holes in main memory | Operating system must maintain a list of free holes in main memory |
| Processor uses page number, offset to calculate absolute address | Processor uses page number, offset to calculate absolute address | Processor uses segment number, offset to calculate absolute address | Processor uses segment number, offset to calculate absolute address |
| All the pages of a process must be in main memory for process to run, unless overlays are used | Not all pages of a process need be in main memory frames for the process to run. Pages may be read in as needed | All the segments of a process must be in main memory for process to run, unless overlays are used | Not all segments of a process need be in main memory for the process to run. Segments may be read in as needed |
| | Reading a page into main memory may require writing a page out to disk | | Reading a segment into main memory may require writing one or more segments out to disk |

Operating Systems: Internals and Design Principles, 9th Edition, by Williams Stallings.