Virtual Memory in Real Systems

Windows OSes

Handled by Virtual Memory Manager in NT Executive (kernel)

32-bit address space (4GB). Upper 2GB is the same for all processes and used by the kernel. Lower 2GB is process specific.

Page Table Entry

- Page Directory 10 bits
- Page Table 10 bits
- Offset 12 bits (4K page size Pentium. Can be larger).

Supports memory-mapped files.

Virtual Memory Manager

See page table entry, Tanenbaum Fig 11-26.

On page fault bring page into free frame (tries to keep an available list of frames through a *working set manager* running every second).

Fault in a few adjacent pages—anticipating locality of reference

Initial minimum working set size of 20-50 pages.

Can extend beyond this minimum, but if a free frame is needed then working set manager considers large, inactive processes as the first to take pages from when more are needed.

Memory manager also tests working set size by periodically stealing "victim" pages (mark page as not present). If no page fault occurs because of this theft then reduce the working set size by one.

Linux

3 GB of address space for process, 1GB for page table and kernel data.

4KB page size on pentium, 8KB page size on alpha

Uses 3-level page tables (Fig 10-17).

Demand-paging.

Virtual Memory Regions

run of consecutive pages wiht same protection/paging properties

- paging disk (swap area)
- file (text segment, memory-mapped files)

Supports memory mapped files.

Uses buddy algorithm for supporting dynamically loaded modules (e.g. device drivers).

New Virtual Memory

fork() copies the page table. Shared references to common pages. Use copy-on-write if a shared page is written to.

exec() creates a new page table.

Page Replacement Algorithm

Variation on the clock algorithm with a paging daemon.