# Virtual Memory – Demand Paging

When a reference is made to an address on a page not present in main memory, a **page fault**, is said to occur.

The operating system must read in the required page into memory from the disk, enter its new physical memory location in the page table, and then repeat the instruction that caused the fault.

This method of obtaining a virtual memory is called **demand paging:** a page is brought into memory only when a request for it occurs, not in advance.

231

# Virtual Memory – Working Set Model

Most programs do not reference their address space uniformly - references tend to cluster on a small number of pages.

This concept is called the **locality principle**.

A memory reference may fetch an instruction, it may fetch data, or it may store data. At any instant in time, *t*, there exists a set consisting of all the pages used by the *k* most recent memory references. This is called the **working set**.

Because the working set normally varies slowly with time, it is possible to make a reasonable guess at the pages that will be needed when the program is started/restarted, on the basis of the working set when it was last stopped. These pages can then be loaded in advance before starting the program up (assuming they fit).

232

# Page Replacement Policy

Programmers rarely know which pages are in the working set, so the operating system must discover this set dynamically.

To make room for a new page, some other page will generally have to be sent back to the disk. Thus, an algorithm is needed to decide which page to remove.

Most operating systems try to predict which of the pages in memory is the least useful, in the sense that its absence would have the smallest adverse effect on therunning program.

**Least Recently Used (LRU) Algorithm**

Evicts the page **least recently used** because the probability of its not being in the current working set is high.

233

# Page Replacement Policy

**First-In First-Out (FIFO) Algorithm**

- FIFO removes **the oldest page**, independently of when this page was last referenced.
- Associated with each page frame is a counter.
    - Initially, all the counters are set to 0.
    - After each page fault has been handled, the counter for each page present in memory is incremented
    - The counter for a page brought into memory is initialized to 0.

- In choosing a page to remove, the page with the highest counter value is chosen.

Problem with page replacement algorithms:

If the working set is larger than the number of available page frames, no algorithm will give good results, and page faults will be frequent - although, LRU will tend to minimize the number of page faults.

234

# Page Replacement Policy

**Thrashing**
A program that generates page faults frequently and continuously is said to be **thrashing**.

If a program, even one using a large virtual address space, has a small, slowly changing working set that fits in available main memory, it will perform well.

If a page about to be evicted has not been modified since it was read in (a likely occurrence if the page contains program rather than data), it is not necessary to write it back onto disk. If it has been modified since it was read in, the copy on the disk will be out of date, and the page must be written to the disk.

235