Chapter 10 Questions

Questions may be asked in class, on quizzes, and on exams

* + Briefly describe the value of implementing virtual memory in an OS
  + **Virtual memory** is a **memory management** capability of an **operating system** (**OS**) that uses hardware and software to allow a computer to compensate for physical**memory** shortages by temporarily transferring data from random access **memory**(RAM) to disk storage
  + Briefly explain the difference between demand paging and prepaging; explain why one of them might be better than the other

**demand paging**, a page is brought into memory only when a location on that page is actually referenced during execution. pages are loaded only when

they are ***demanded*** during program execution

* + **pre**-**paging**, pages other than the one demanded by a page fault are brought in. The selection of such pages is done based on common access patterns, especially for secondary memory devices.
  + demand paging does not require the entire process (all pages) to be in memory before the program execution begins. They can be loaded off the disk as and when required. A page fault triggers this
  + Explain how a parent and child process may share data, and how the copy-on-write action keeps this from becoming a problem
  + In Unix-like systems to create a new process fork() system call is used. The process that calls fork() is the parent process and the newly created process is its child
  + now there are two processes and they must have separate address space. But in Linux a technique called copy-on-write is used, due to this the parent and child share a single copy of the process address space.

**copy-on-write**, which works by allowing the parent and child processes initially to share the same

pages.

* + Explain how a working set is used and what changes in a working set improve and/or cause

The **working-set model** is based on the assumption of locality. This model

uses a parameter, Δ, to define the **working-set window**. The idea is to examine

the most recent Δ page references. The set of pages in the most recent Δ page

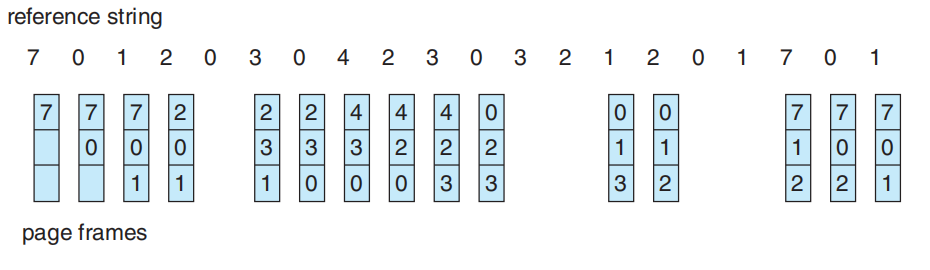
references is the **working set** (Figure 10.22). If a page is in active use, it will be in

the working set. If it is no longer being used, it will drop from the working set

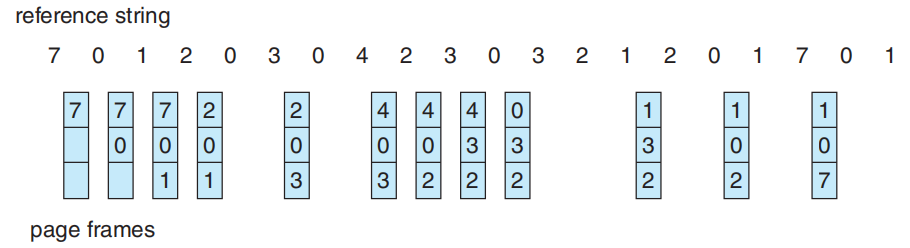
Δ time units after its last reference. Thus, the working set is an approximation

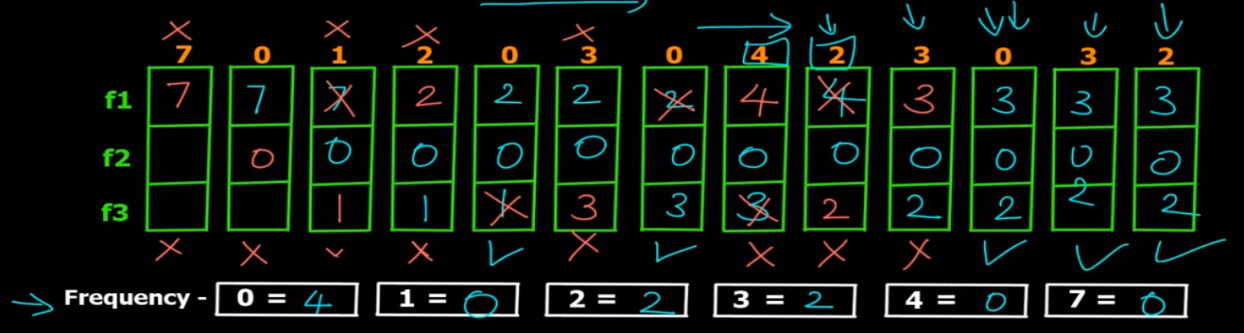
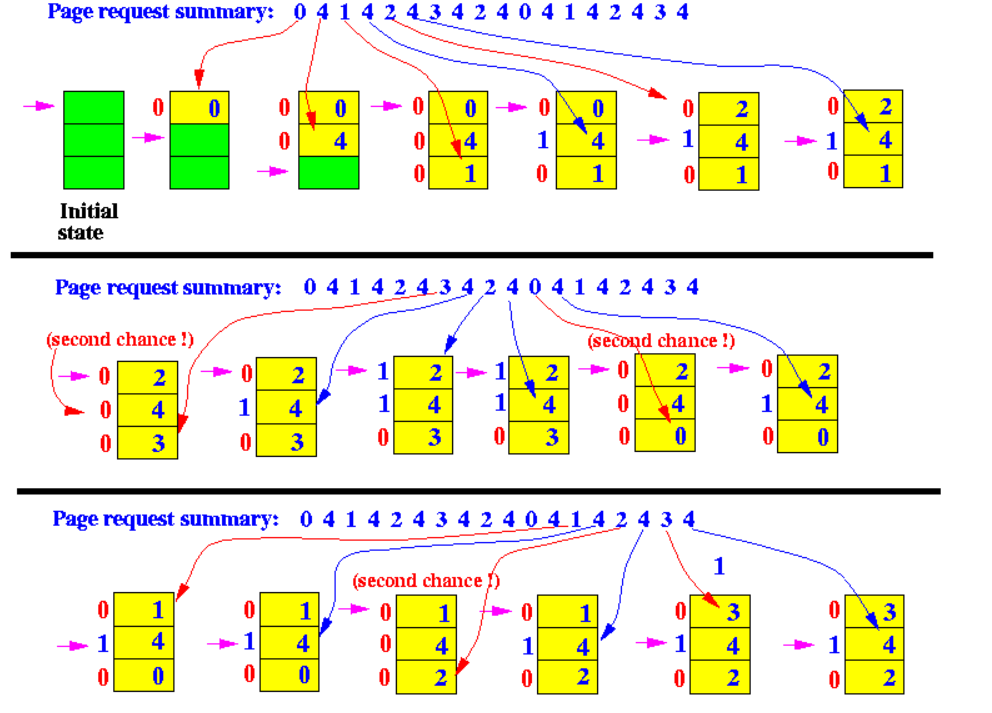
of the program’s locality

* + Given a working reference string, show how the FIFO page replacement would manage page replacement, and identify any possible logistical issues



* + Given a working reference string, show how the LRU page replacement would manage page replacement, and identify any possible logistical issues



* + Given a working reference string, show how the LFU page replacement would manage page replacement, and identify any possible logistical issues
  + 
  + Given a working reference string, show how the Second-Chance or Clock page replacement would manage page replacement, and identify any possible logistical issues
  + 
  + Explain thrashing and how it impacts system operation.

if a process does not have “enough” frames— that is, it

does not have the minimum number of frames it needs to support pages in the

working set. The process will quickly page-fault.it must replace

some page. Consequently, it quickly faults again, and

again, and again, replacing pages that it must bring back in immediately.

This high paging activity is called **thrashing**.

**thrashing** occurs when a computer's virtual memory resources are overused,

Leading to a constant state of paging and page faults, inhibiting most application-level processing.

This causes the performance of the computer to degrade or collapse.

* + Analyze the plot at Figure 10.20 and explain system dynamics at specified points over the curve.
  + 10.6.2
  + Explain physical locality and how it compares to temporal locality.

In **physics**, the principle of **locality** states that an object is directly influenced only by its immediate surroundings.

**Temporal locality** refers to the reuse of specific data, and/or resources, within a relatively small time duration。 How often get used => frequence

* + End of Chapter 10
    - * Exercises: 10.5, 10.8, 10.9, 10.17, 10.19, 10.24, 10.26, 10.27