Announcements

- Reading Chapter 10
- Midterm #1
  - Last day to submit requests is Thursday

Inverted Page Tables

- Solution to the page table size problem
- One entry per page frame of physical memory

  <process-id, page-number>

  – each entry lists process associated with the page and the page number

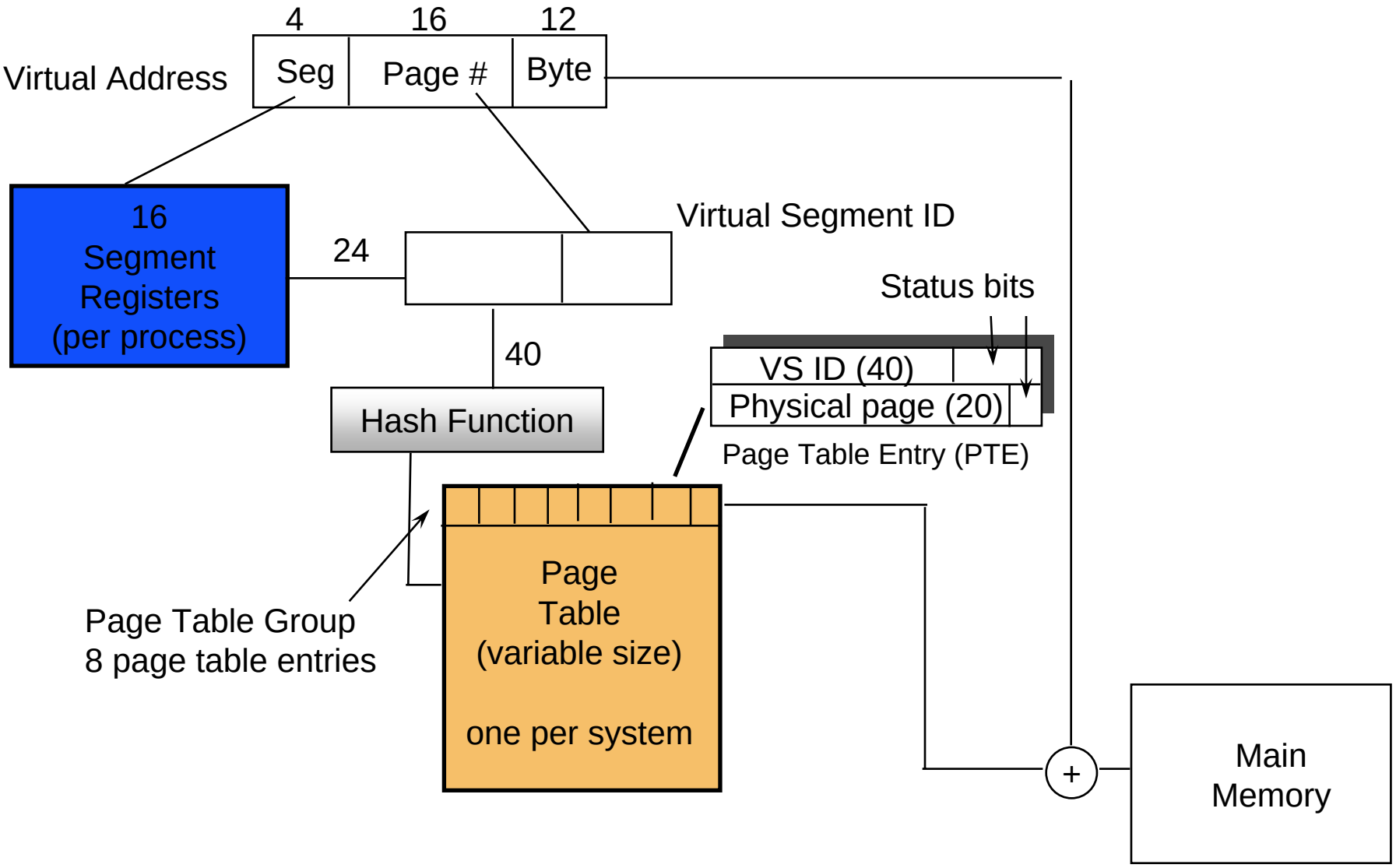
  – when a memory reference:

    - **<process-id,page-number,offset>**occurs, the inverted page table is searched (usually with the help of a hashing mechanism)
    - if a match is found in entry ***i*** in the inverted page table, the physical address **<i,offset>** is generated

  – The inverted page table does not store information about pages that are not in memory

    - page tables are used to maintain this information
    - page table need only be consulted when a page is brought in from disk

Inverted Page Table Example (PPC)



File Abstraction

- What is a file?
  - A named collection of information stored on secondary storage
- Properties of a file
  - non-volatile
  - can read, read, or update it
  - has meta-data to describe attributes of the file
- File Attributes
  - name: a way to describe the file
  - type: some information about what is stored in the file
  - location: how to find the file on disk
  - size: number of bytes
  - protection: access control
    - may be different for read, write, execute, append, etc.
  - time: access, modification, creation
  - version: how many times has the file changed

File Operations

- Files are an abstract data type
  - interface (this lecture)
  - implementation (next lecture)
- create a file
  - assign it a name
  - check permissions
- open
  - check permissions
  - check that the file exists
  - lock the file (if we don't what to permit other users a the same time)

File Operations (cont)

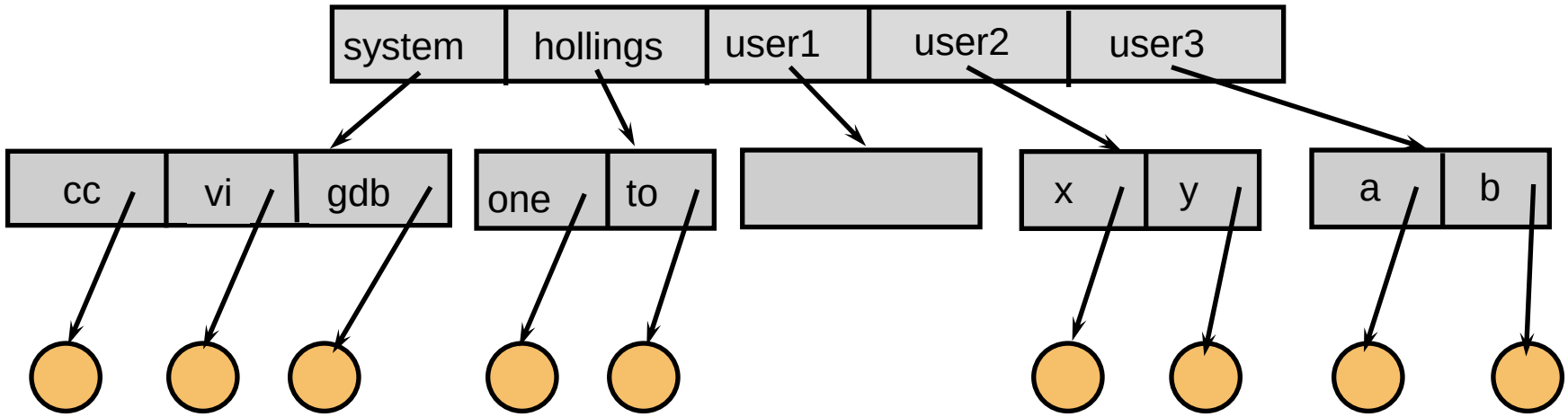
- write
  - indicate what file to write (either name of handle)
  - provide data to write
  - specify where to write the data within the file
    - generally this is implicit (file pointer)
    - could be explicit (direct access)
- read
  - indicate what file to read (either name of handle)
  - provide place to put information read
  - indicate how much to read
  - specify where to write the data within the file
    - generally this is implicit (file pointer)
    - could be explicit (direct access)
- fsync (synchronize disk version with in-core version)
  - ensure any previous writes to the file are stored on disk

File Operations (cont)

- seek
  - move the implicit file pointer to a new offset in the file
- delete
  - remove named file
- truncate
  - remove the data in the file from the current position to end
- close
  - unlock the file (if open locked it)
  - update meta data about time
  - free system resources (file descriptors, buffers)
- read meta data
  - get file size, time, owner, etc.
- update meta data
  - change file size, time owner, etc.

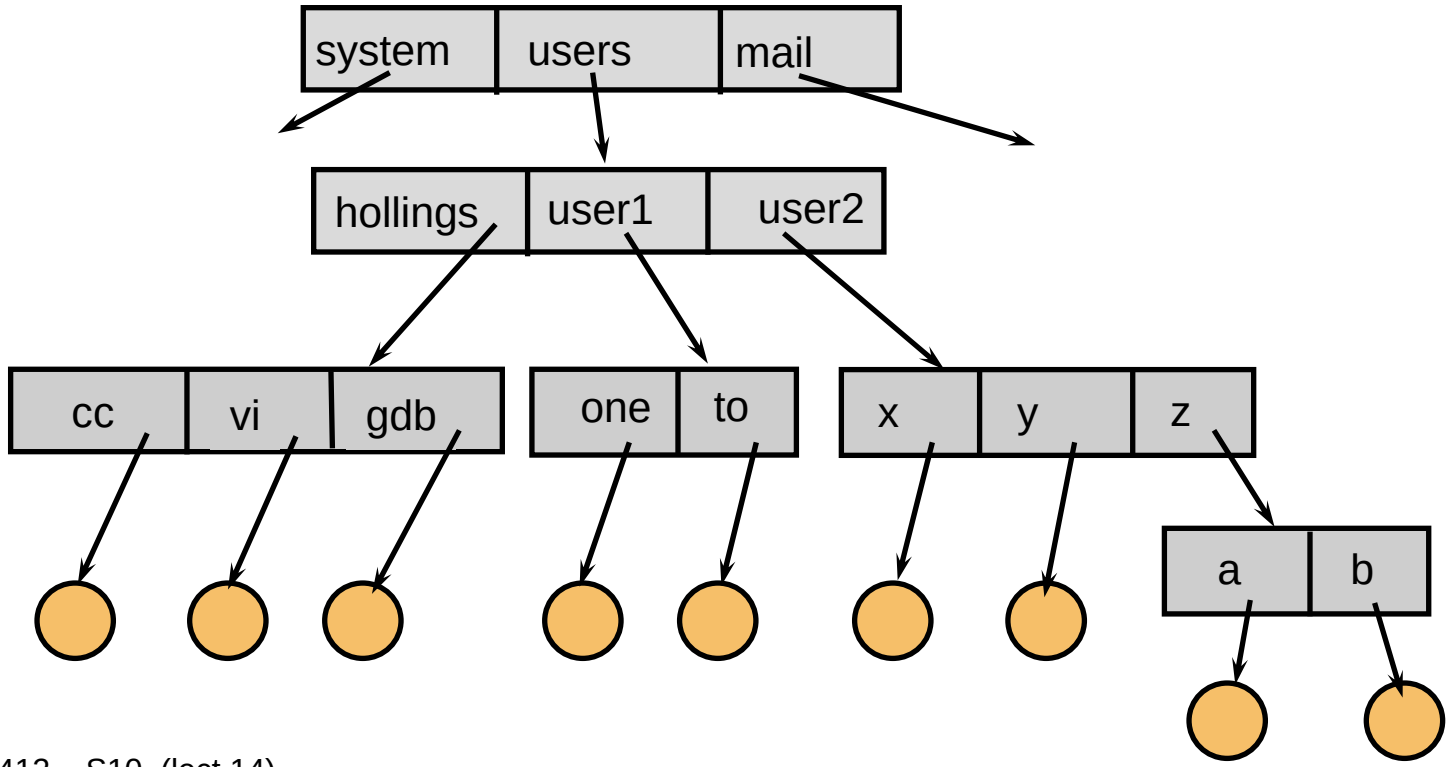
Simple Directory Structures

- One directory
  - Having all of the files in one name space is awkward
  - lots of files to sort through
  - different users would have to coordinate file names
  - each file has to have a unique name
- Two level directory
  - top level is users
  - second level is files per user



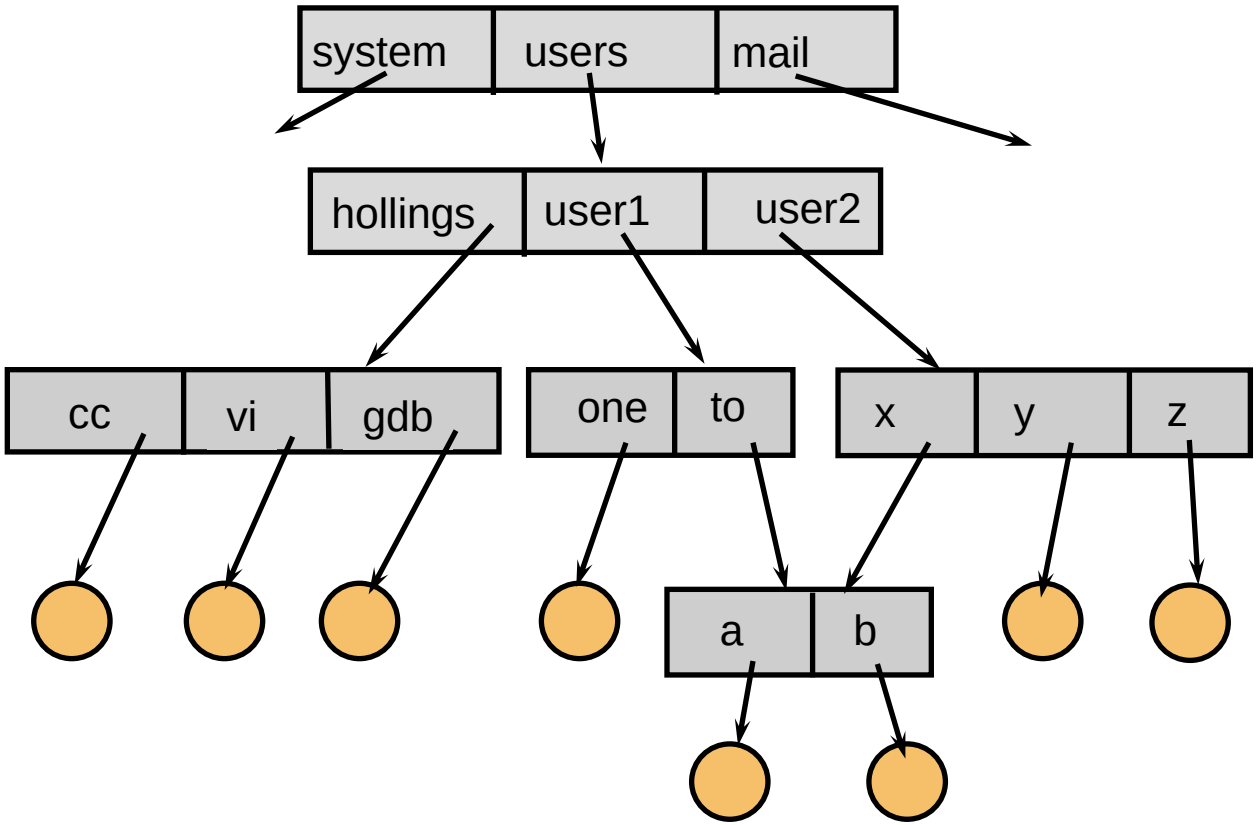
Tree Directories

- create a tree of files
- each directory can contain files or directory entries
- each process has a current directory
  - can name files relative to that directory
  - can change directories as needed



Acylic Graph Directories

- Permit users to share subdirectories



Issues for Acylic Graph Directories

- Same file may have several names
  - absolute path name is different, but the file is the same
  - similar to variable aliases in programming languages
- Deletion
  - if one user deletes a file does it vanish for other users?
    - yes, it should since the directory is shared
  - what if one user deletes their entry for the shared directory
    - no, only the last user to delete it should delete it
    - maintain a reference count to the file
- Programs to walk the DAG need to be aware
  - disk usage utilities
  - backup utilities