Today

• Page replacement algorithms
Virtual Memory

- Each process has the illusion of a large contiguous address space.
- However, physical memory might be much smaller than the sum of the memory request of the ready processes.
- How do we give this illusion to multiple processes?
  - With Virtual Memory, some frames may reside in disk.
Virtual Memory

Virtual memory

Physical memory

page table

disk
Valid bit

Extend page table entry with a valid bit:

- If page in memory, the valid bit is set to $v$, otherwise, the valid bit is set to $i$.
- If page is in memory, translation works as before.
- If page is not in memory, translation causes a **page fault**.
Page Fault

On a page fault:

- OS finds a **free frame**, or evicts one from memory.
- Issues disk request to **fetch data** for page.
- **Block** current process, **context switch** to new process.
- When disk completes, set **valid bit to v**, and current process in ready queue.
Page Fault

1. Trap
2. Page is on backing store
3. Bring in missing page
4. Reset page table
5. Free frame
6. Restart instruction
Page Replacement

• When a process has used up all frames it is allowed to use, OS must select a page to eject from memory to allow new page.

• The page to eject is selected using a Page Replacement Algorithm.
Page Replacement

1. Swap out victim page
2. Change to invalid
3. Swap desired page in
4. Reset page table for new page
Dirty Bits

- Use **dirty bit** to reduce **overhead of page transfers**.
- Only modified pages are written to disk.
- Non-modified pages can always be brought back from the original source.
  - Program code segments are rarely modified, can bring pages back from the program image stored on disk
Page Replacement Algorithms

- **FIFO**: the page brought in earliest is evicted
- **OPT**: evict page that will not be used for longest period of time
- **LRU**: evict page that has not been used the longest
- **MRU**: evict the most recently used page
- **LFU**: evict least frequently used page
FIFO

- A FIFO queue holds all pages in memory.
- The OS replaces the page at the head of the queue.
- A newly brought page is placed at the tail of the queue.
FIFO

- Reference string: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5
- 3 frames (3 pages can be in memory at a time per process):

  1  4  5
  2  1  3  9 page faults
  3  2  4

- 4 frames:

  1  5  4
  2  1  5  10 page faults
  3  2
  4  3

- Belady’s Anomaly: sometimes, more frames → more page faults
Belady’s Anomaly

![Graph showing Belady’s Anomaly](image)
Optimal Algorithm

- Replace page that will not be used for longest period of time.
- 4 frames example
- 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5

```
  1   4
  2   6 page faults
  3
  4   5
```

- Used for measuring how well an algorithm performs.
OPT Approximation

- In real life, we do not have access to the future page request stream of a program.
- So we need to make a best guess for which pages will not be used for the longest time.
Today

• Page replacement algorithms
Coming up...

- Next lecture: more page replacement algorithms
- HW3 is due on Monday
- Next in-class exam on Wednesday