CIS 452 - Operating Systems Concepts

Nathan Bowman

Images taken from Silberschatz book

FIFO Replacement

Due to over-allocation of memory in virtual memory, sometimes system must replace a page to make room for another

Effective page replacement algorithm can have large impact on system performance

What makes a good page-replacement algorithm?

Recall that expected access time (EAT) depends heavily on page fault rate

EAT = (1 - p)*ma* + *p*(page fault time)

OS designers do not have control over memory access time (ma) and have little control over page fault time

Goal: minimize page fault rate

Difficult to say anything mathematically about different page-replacement algorithms

Instead, compare on examples of memory accesses

Memory accesses can be:

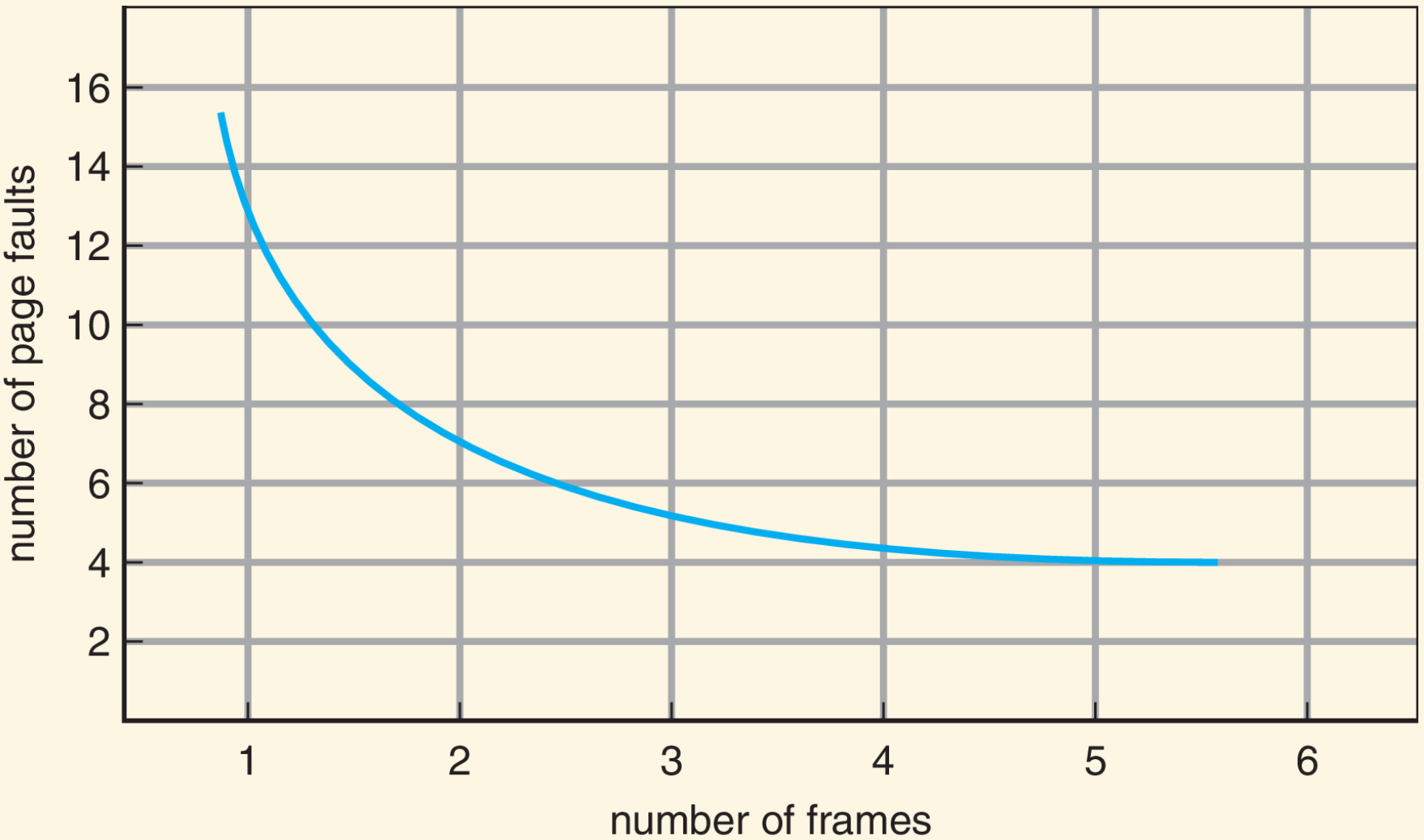
- randomly generated
- based on recording actual system

Either way, concerned only with which page was accessed, not which offset within page

To perform evaluation, need to know how much memory is in system

Easy way to lower page fault rate (usually) is to increase memory

Our examples assume system has three frames

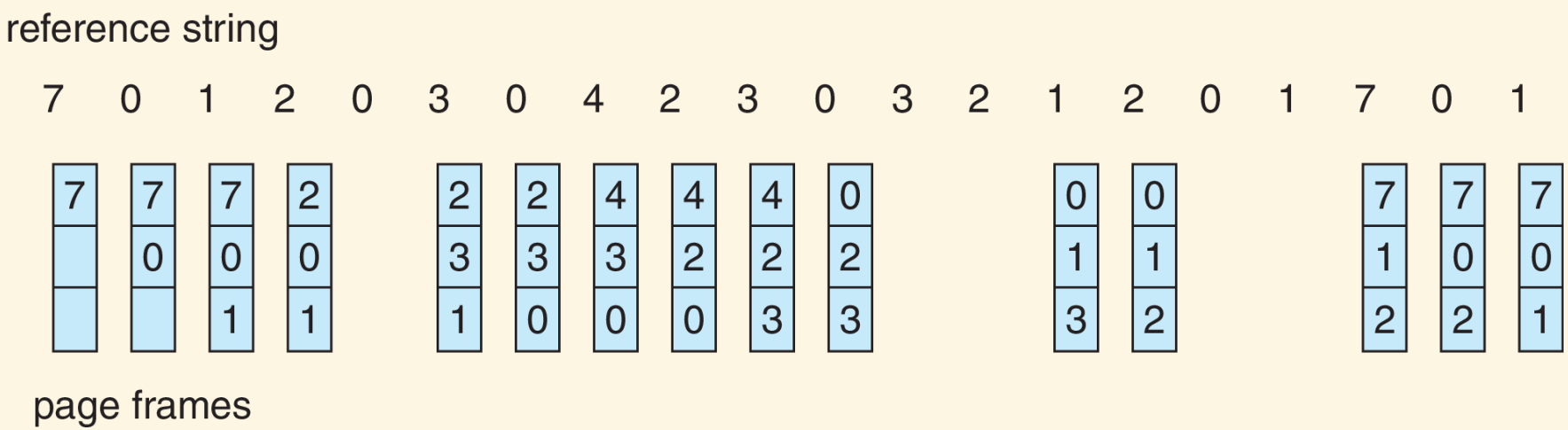

number of page faults
16
14
12
10
8
6
4
2
1
2
3
4
5
6
number of frames

First-In First-Out (FIFO) replacement always removes page that has been in memory longest

Process order usually tracked by maintaining queue rather than recording actual entry times

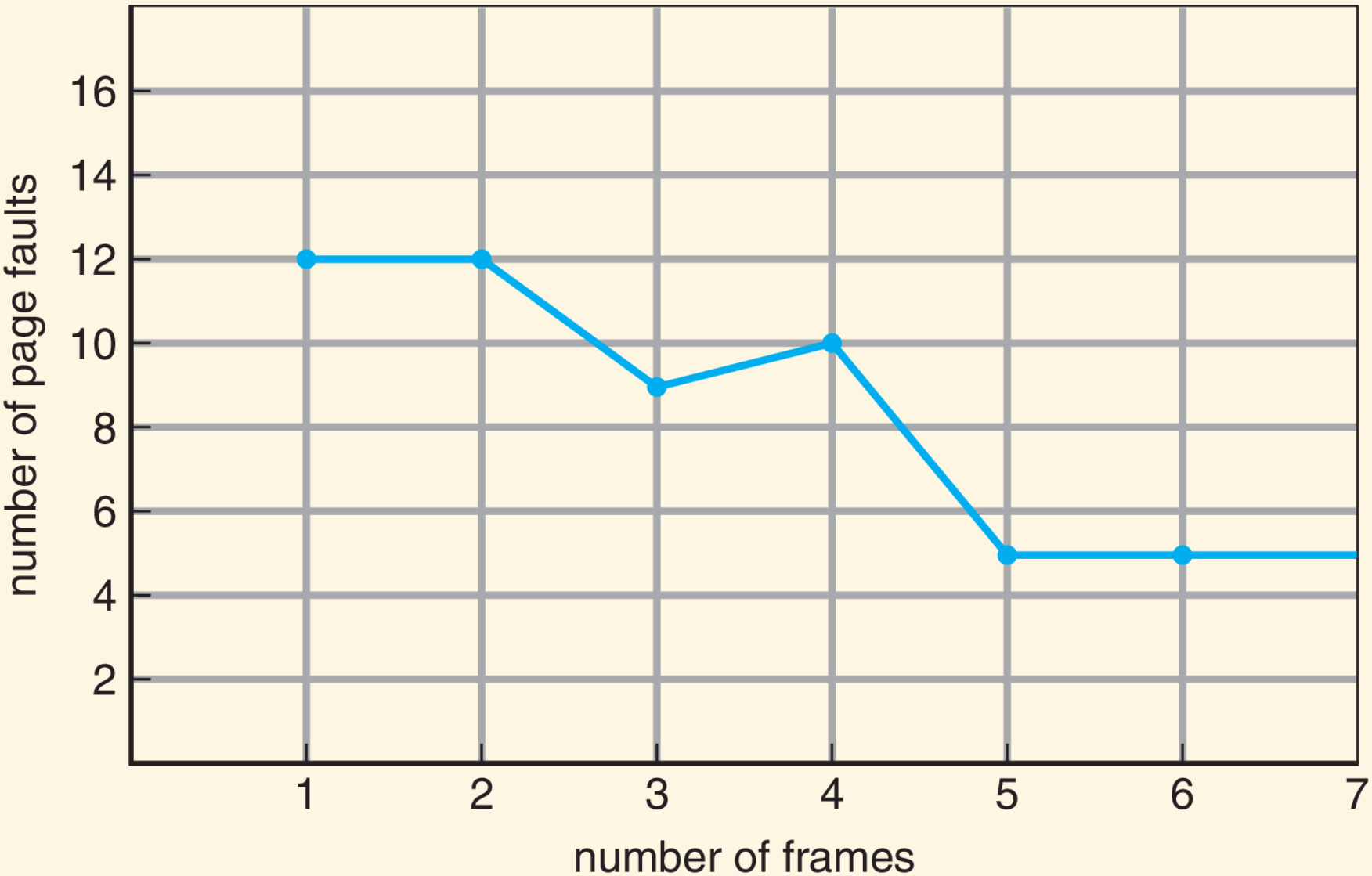
Simple to implement

Does not account for use -- page that is frequently used will still be kicked out of memory if it is oldest


reference string
7 0 1 2 0 3 0 4 2 3 0 3 2 1 2 0 1 7 0 1
page frames

Susceptible to **Belady's anomaly** -- increasing number of frames can *increase* page fault rate

Not generally the trend, but not a good property to have in replacement algorithm


number of page faults
16
14
12
10
8
6
4
2
1
2
3
4
5
6
7
number of frames

Differing page-replacement algorithms affect efficiency, not correctness

If something is paged out that will be needed soon, it will be paged back in when needed

Bad choice of victim page slows down system, but process runs correctly regardless