Systems 3

Memory Management II

Marcel Waldvogel

(Handout)

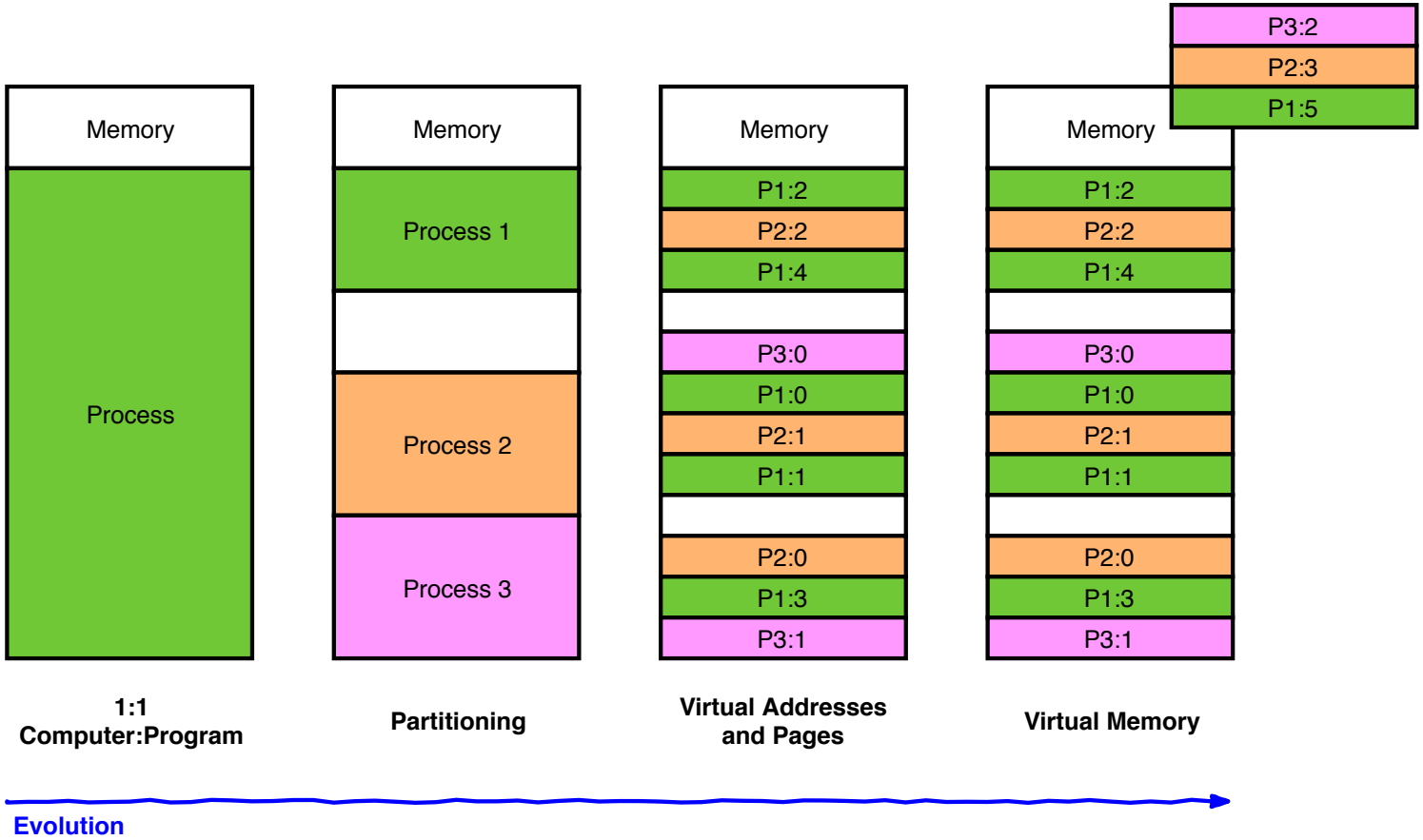
Department of Computer and Information Science
University of Konstanz

Winter 2019/2020

Chapter Goals

- Going beyond physical memory
- How to simplify memory management for the kernel?
- How can this be represented with the existing page table structure?
- How to select which pages to replace on memory pressure?
- What are the (logical) segments of a program?

Memory development



Swapping Entire process image to disk (*Partitioning*, . . .).

(Demand) Paging Pages not yet/recently needed on disk.

Present Bit

Index	Present	Modified	Frame / Info
0	1	1	1234
1	1	0	2600
2	0	-	File #123, block 883
3	0	-	File #123, block 884
4	1	0	1536
5	0	-	Really invalid
...	...	...	...

Overcoming Memory Limits

Approach	Mechanism
Layout	Increase
Static	Swapping
Paged (virtual memory)	Paging

Which page should be replaced?

The three most simple replacement algorithms are:

- Optimal replacement
- Not recently used (NRU) replacement
- First-in, first-out (FIFO) replacement

But there are more algorithms. See next slides.

Second Chance Replacement

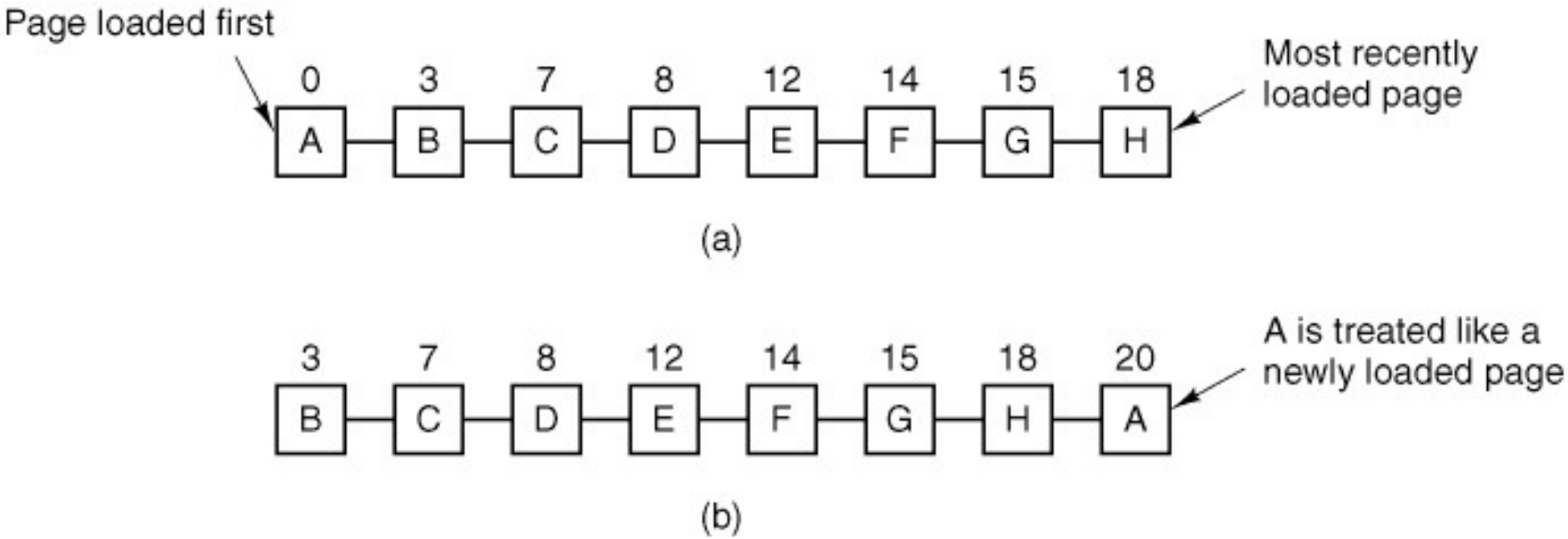


Figure: Operation of second chance. (a) Pages sorted in FIFO order. (b) Page list if a page fault occurs at time 20 and A has its R bit set. The numbers above the pages are their loading times. (Tannenbaum fig. 4-14)

Clock Page Replacement

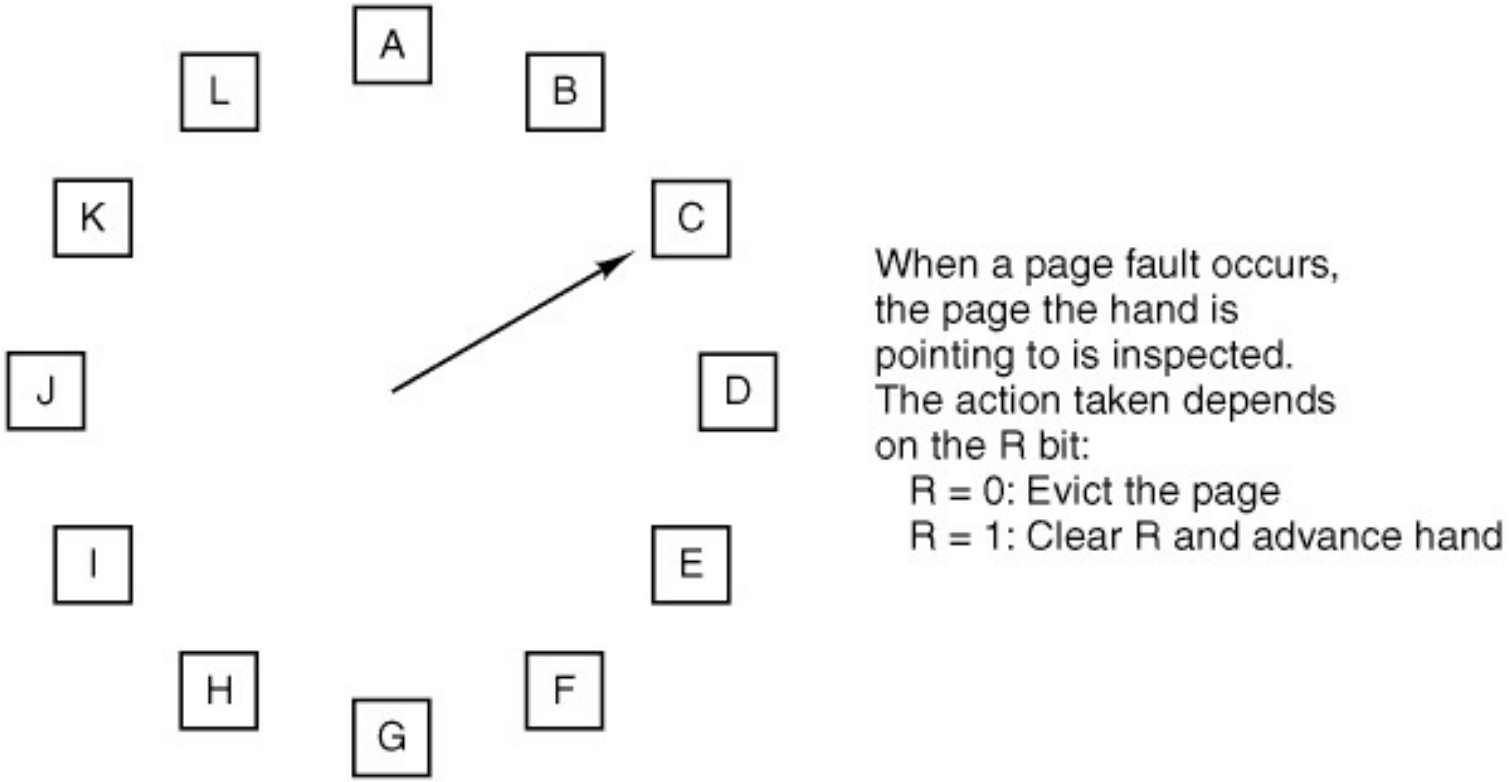


Figure: The clock page replacement algorithm. (Tannenbaum fig. 4-15)

What is the difference between Second Chance and Clock Page Replacement?

Least Recently Used

Assumption: Pages that have not been used for ages will probably remain unused for a long time.

Can be implemented in

- Hardware
- Software

Advantages? Disadvantages?

Simulating LRU

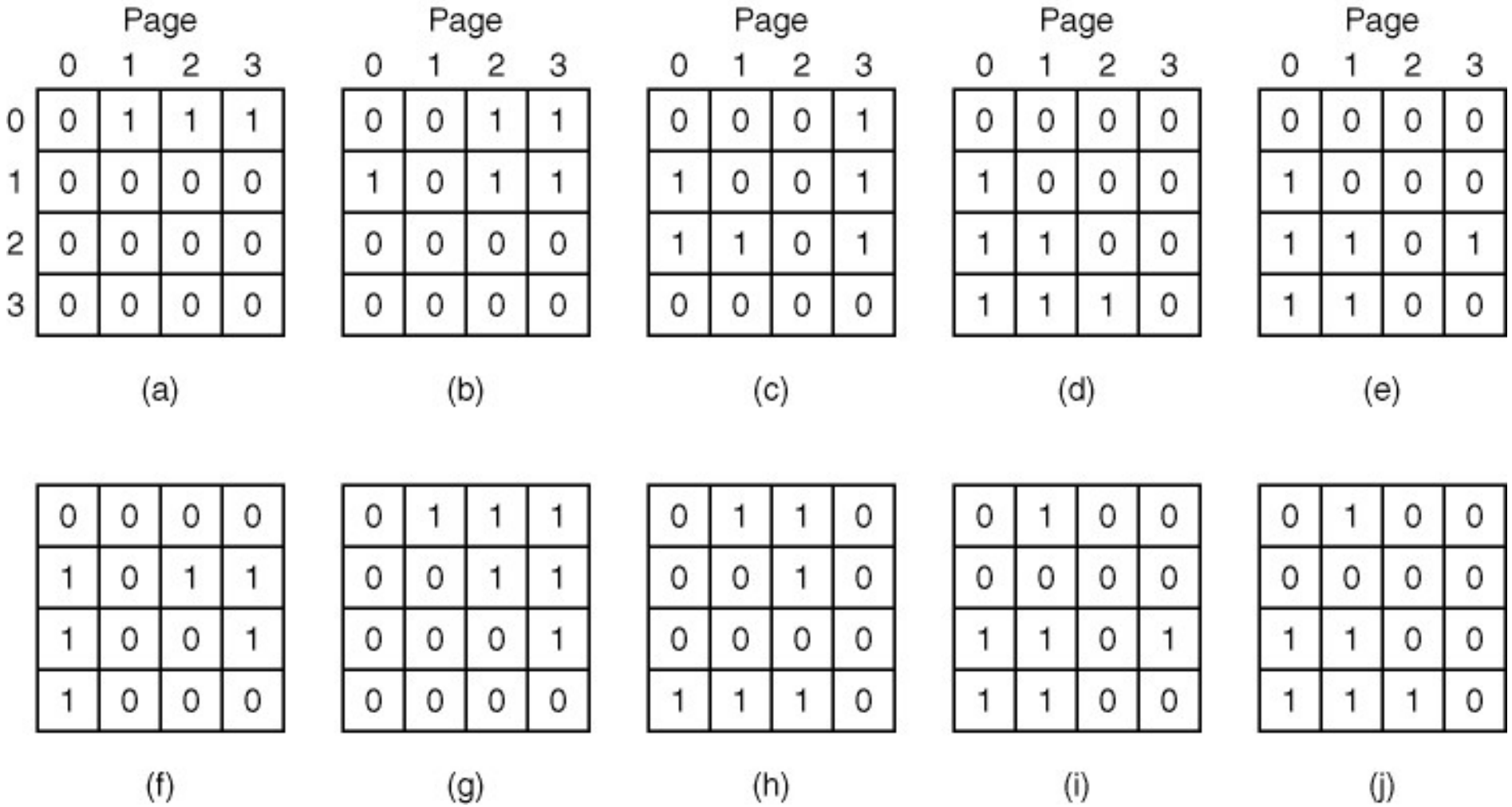
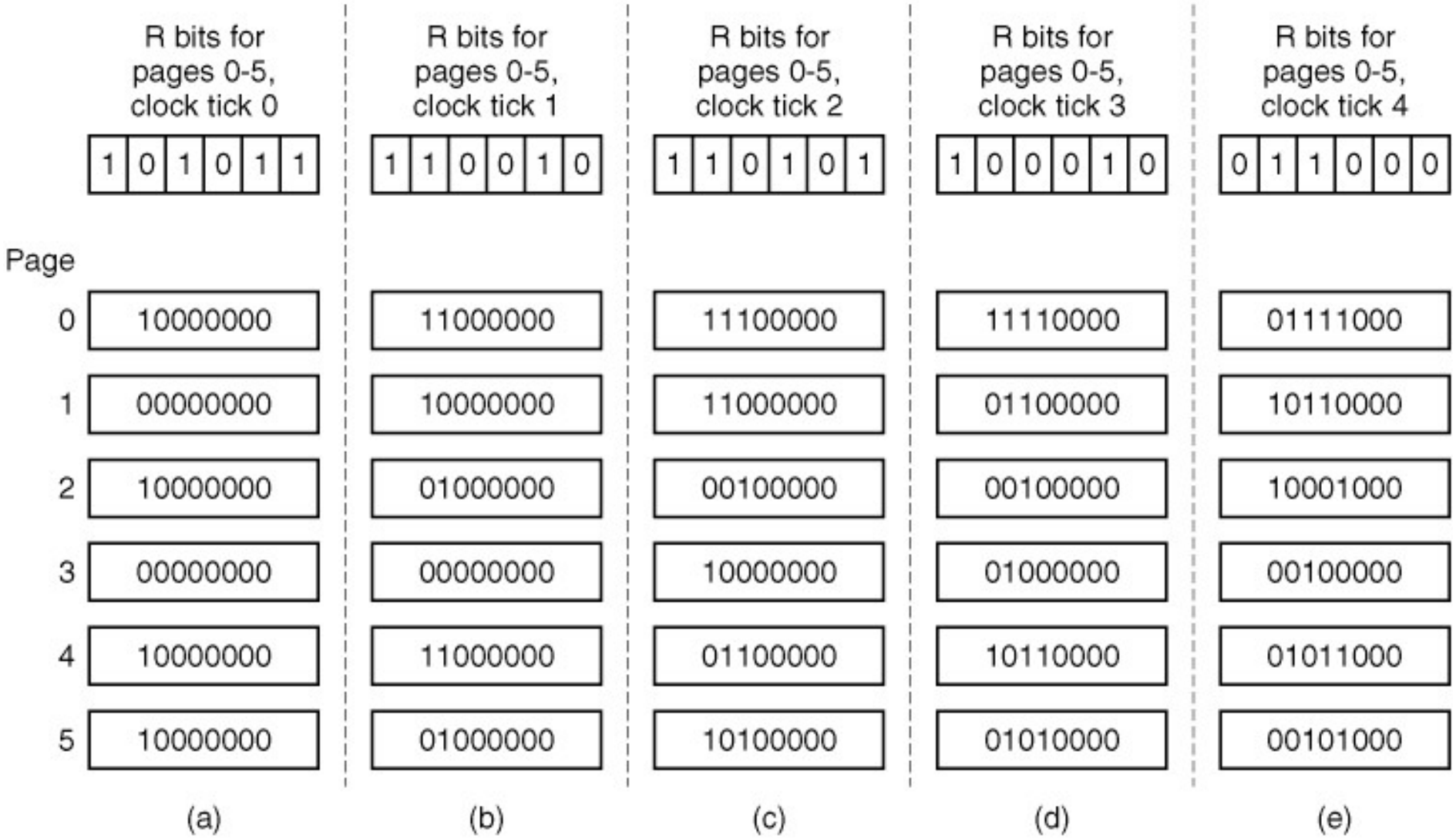


Figure: LRU using a matrix when pages are referenced in order 0, 1, 2, 3, 2, 1, 0, 3, 2, 3. (Tannenbaum fig. 4-16)

Simulating LRU



Working set[1]

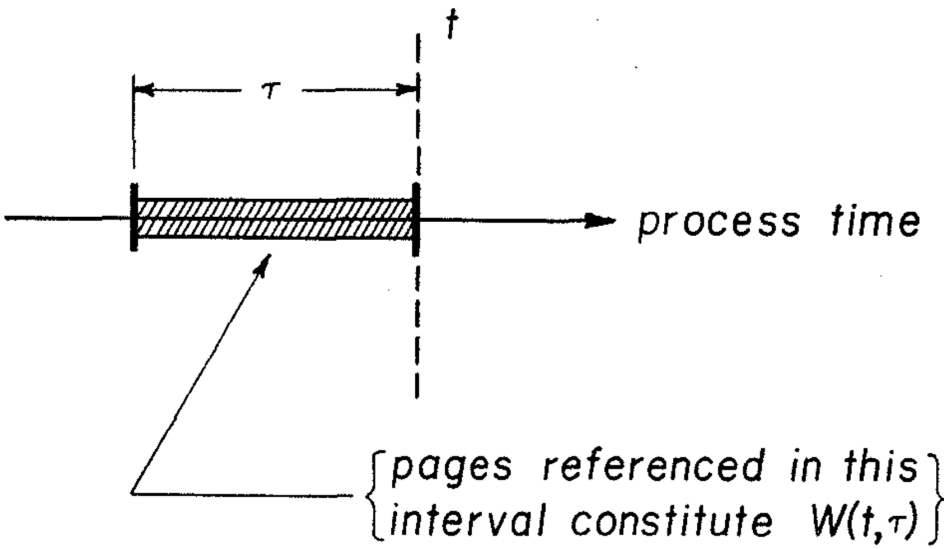


FIG. 2. Definition of $W(t, \tau)$

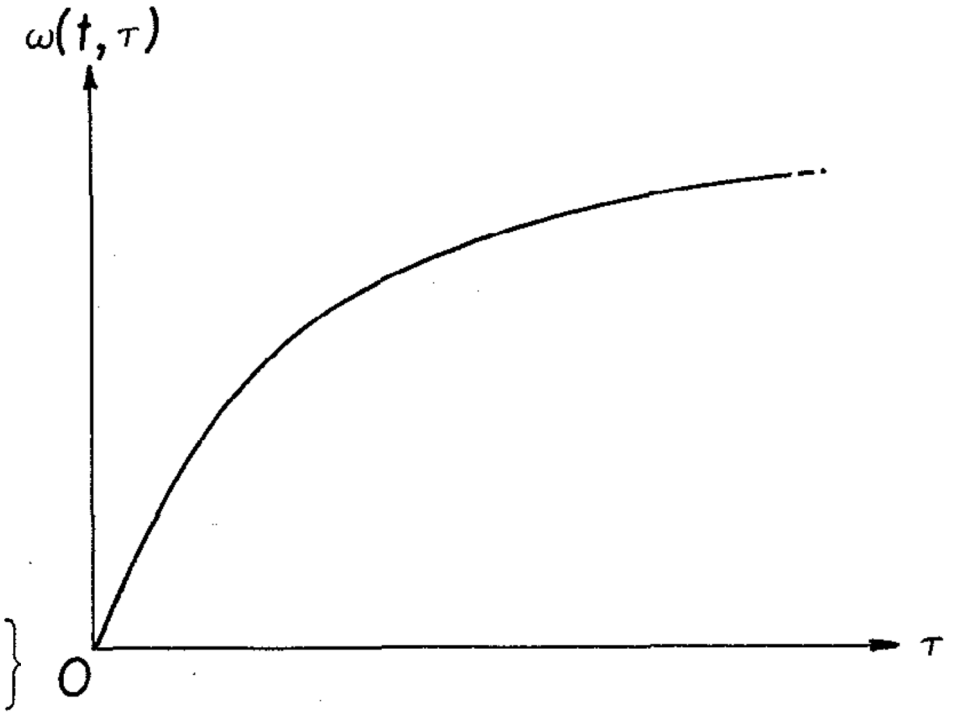


FIG. 3. Behavior of $\omega(t, \tau)$

This locality results in a (slowly) growing working set. Memory areas, which have not been recently used are less likely to be accessed again soon and could be moved to slower memory.

[1] Denning, Peter J. (1968). "The working set model for program behavior". Communications of the ACM. 11(5):323–333

Local vs. Global

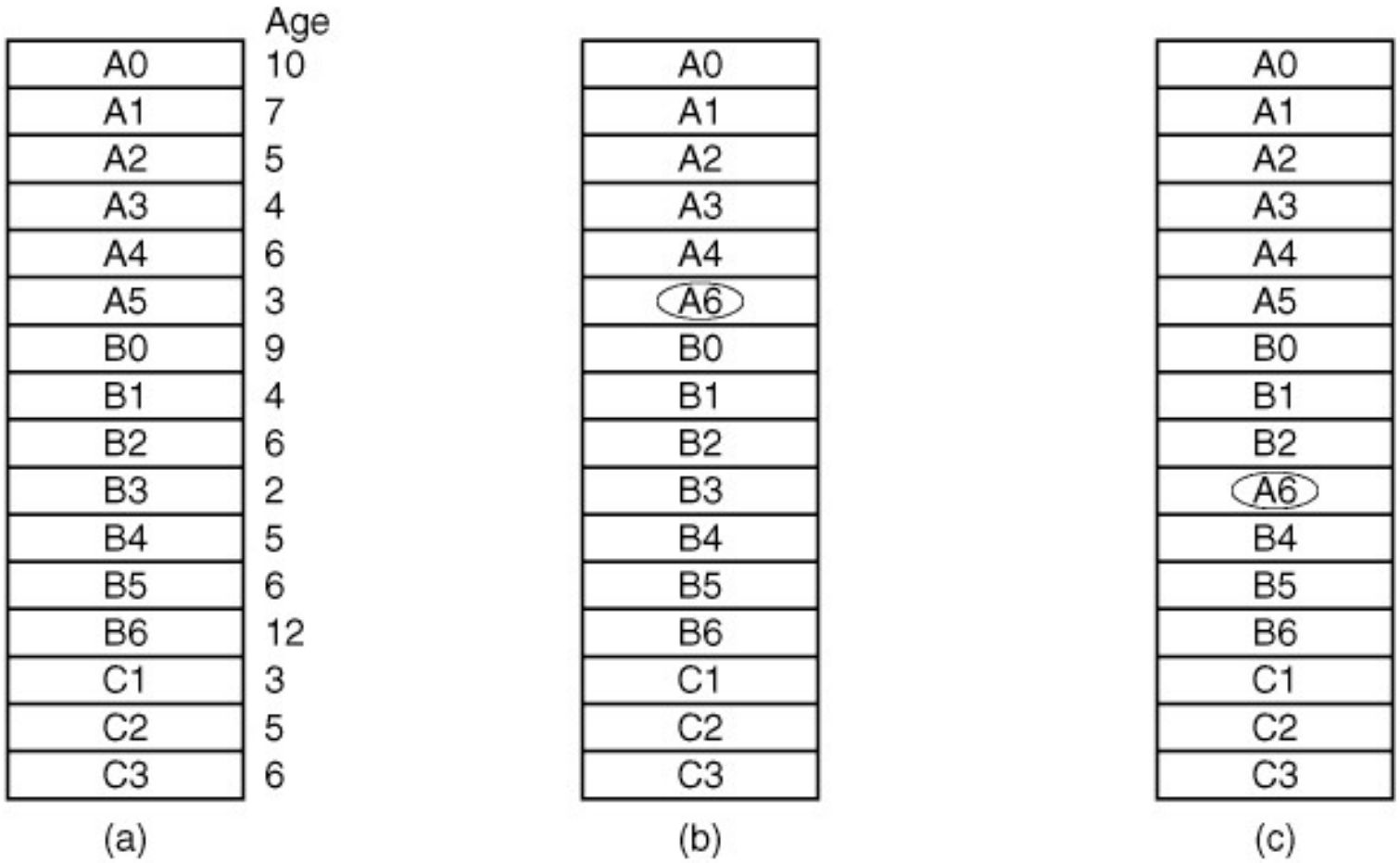


Figure: Local vs. global page replacement. (a) Original configuration. (b) Local page replacement. (c) Global page replacement. (Tannenbaum fig. 4-19)

Page Fault Frequency

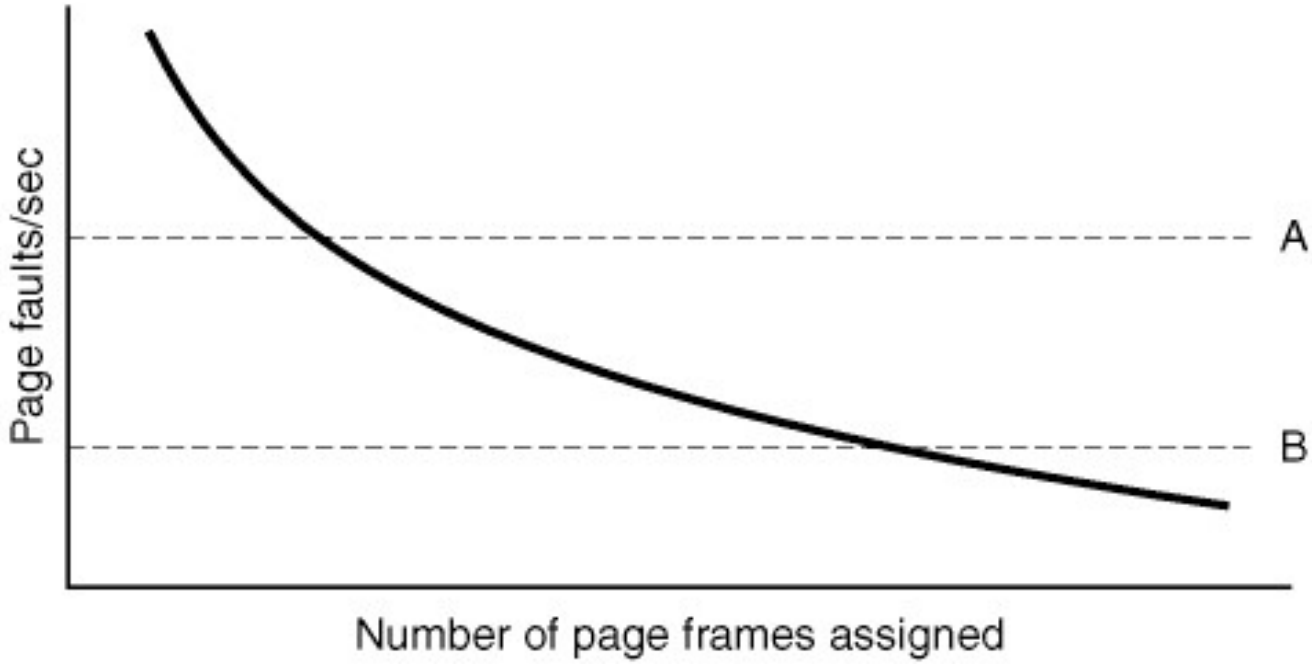


Figure: Page fault rate as a function of the number of page frames assigned. (Tannenbaum fig. 4-20)

Page size

Page size criteria

- What are arguments for large pages?
- What are arguments for small pages?
- What is a useful range?
- Are there applications with different criteria?

Memory Layout of C Programs