Clock-Pro: An Effective Replacement in OS Kernel

Xiaodong Zhang

Ohio State University

Acknowledgement of Contributions:

Song Jiang, Wayne State University

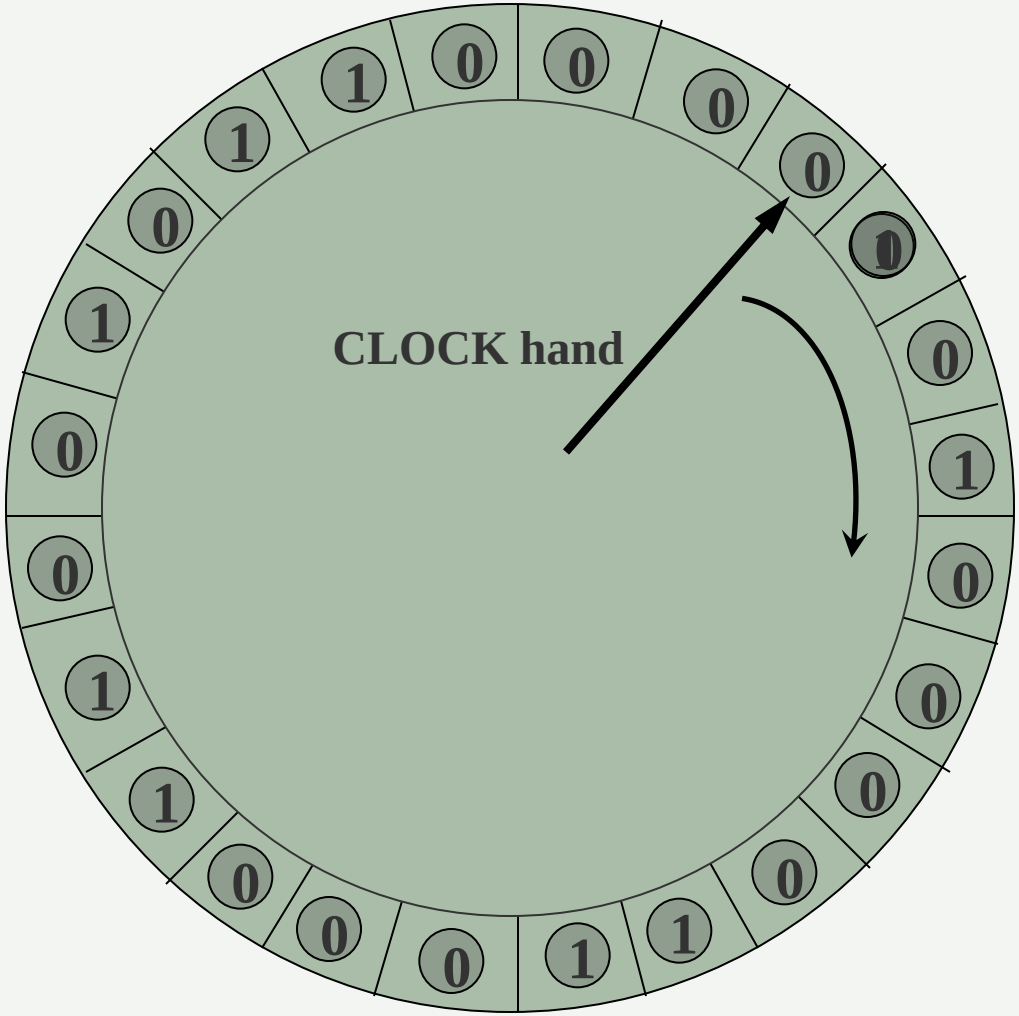
Feng Chen (Ohio State University)

How to Make LIRS Work in OS Kernels?

- Most system kernels use the CLOCK algorithm, an approximation of LRU.
- We have made efforts to directly implement LIRS in Linux kernels, but …
- Our experience tells us that we must build on existing strength.
- CLOCK is the base for LIRS.

Basic CLOCK Replacement

- All the resident pages are placed around a circular list, like a clock;
- Each page is associated with a reference bit, indicating if the page has been accessed.



On a HIT

Set Reference bit to 1
(no algorithm operations)

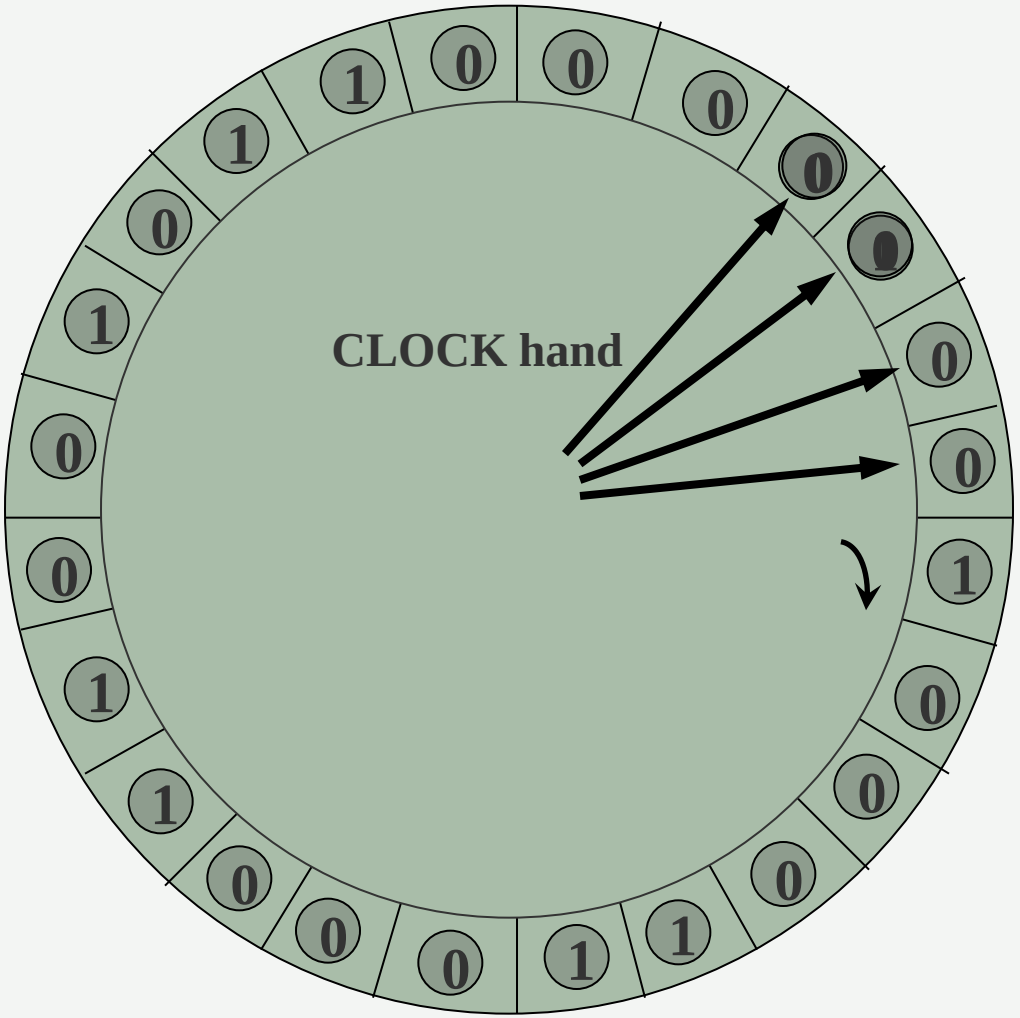
Basic CLOCK Replacement

Starts from the currently pointed page, and evicts the page if it is"`0";

Move the clock hand until reach a "0" page;

Give "1" page a second chance, and reset its "1" to "0"

On a sequence of two MISSes



The CLOCK with a Long History

"In the Multics system a paging algorithm has been developed that has the implementation ease and low overhead of the FIFO strategy and is an approximation to the LRU strategy"

"*A paging Expreiment with the Multics System*" MIT Project MAC Report MAC-M-384, **May 1968**, Fernando J. Corbato (1990 Turing Award Laureate)

The CLOCK has been Widely Used

Major OS

- Multics
- UNIX/AIX/Linux/BSD
- VAX/VMS
- DB2
- Windows/Oracle/Solaris

Major OS Textbooks

- Tanebaum & Woodhull
- Silberschatz & Galvin
- Stallings (for undergraduate)

Prior Work on LRU versus CLOCK

LRU related work

- FBR (1990, SIGMETRICS)
- LRU-2 (1993, SIGMOD)
- 2Q (1994, VLDB)
- SEQ (1997, SIGMETRICS)
- LRFU (1999, OSDI)
- EELRU (1999, SIGMETRICS)
- MQ (2001, USENIX)
- LIRS (2002, SIGMETRICS)
- ARC (2003, FAST)

CLOCK related work

- GCLOCK (1978, ACM TDBS)

1968

2003

- CAR (2004, FAST)
- CLOCK-Pro (2005, USENIX)

GCLOCK Replacement

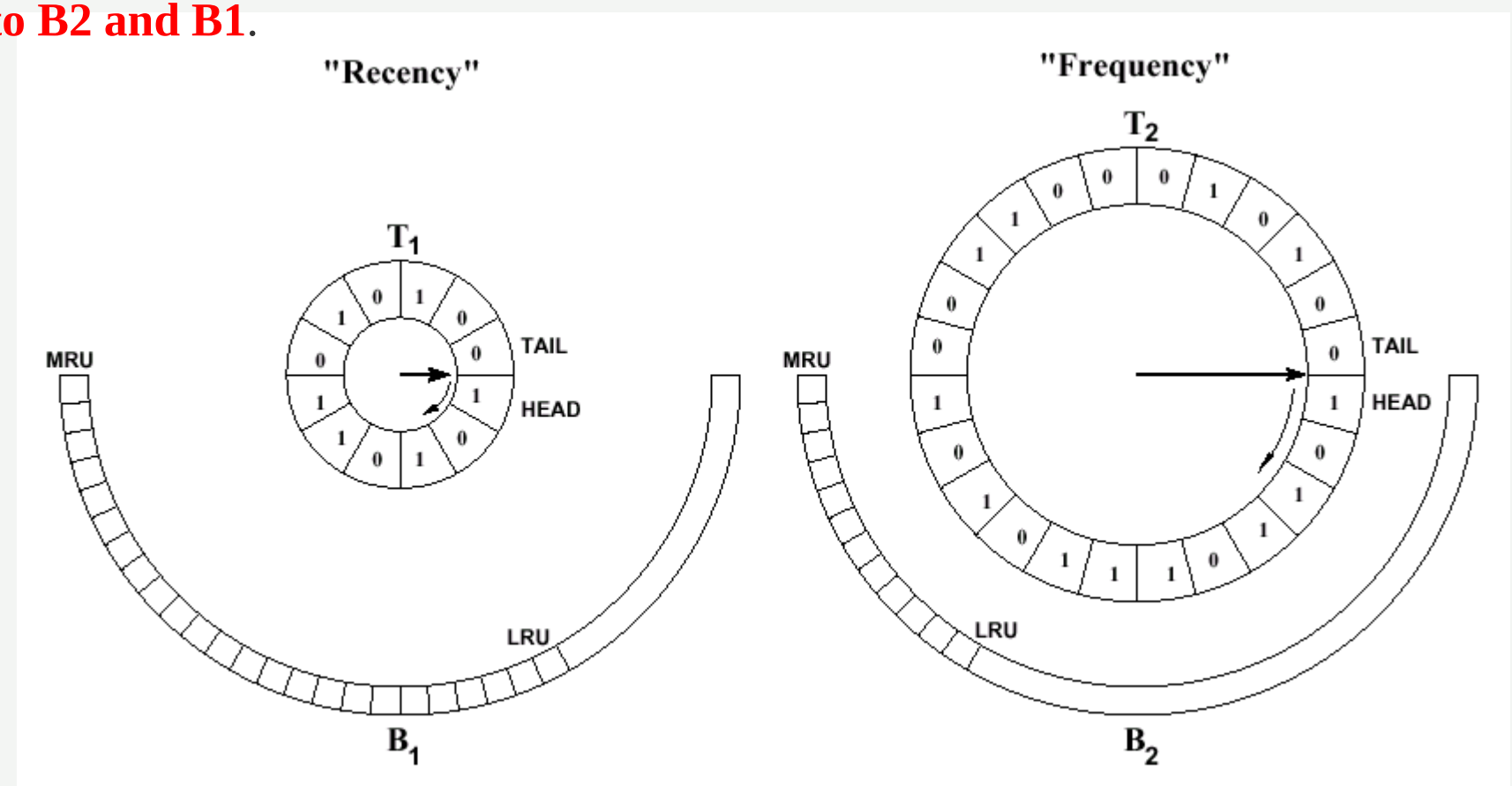
- Introduce additional page access information.
- A **counter** is associated with each page rather than a single bit;
- The counter is **incremented** on a page hit;
- The clock hand periodically moves, and **decrements** the counter of each block;
- The page with its **counter of 0** is replaced.

Age-Based CLOCK in Linux and FreeBSD

- An **age** is associated with each page in addition to a reference bit;
- When the clock hand sweeps through pages, it **increases** its age if the page's bit is 1, otherwise it **decreases** its age.
- The page with its **age of 0** is replaced.

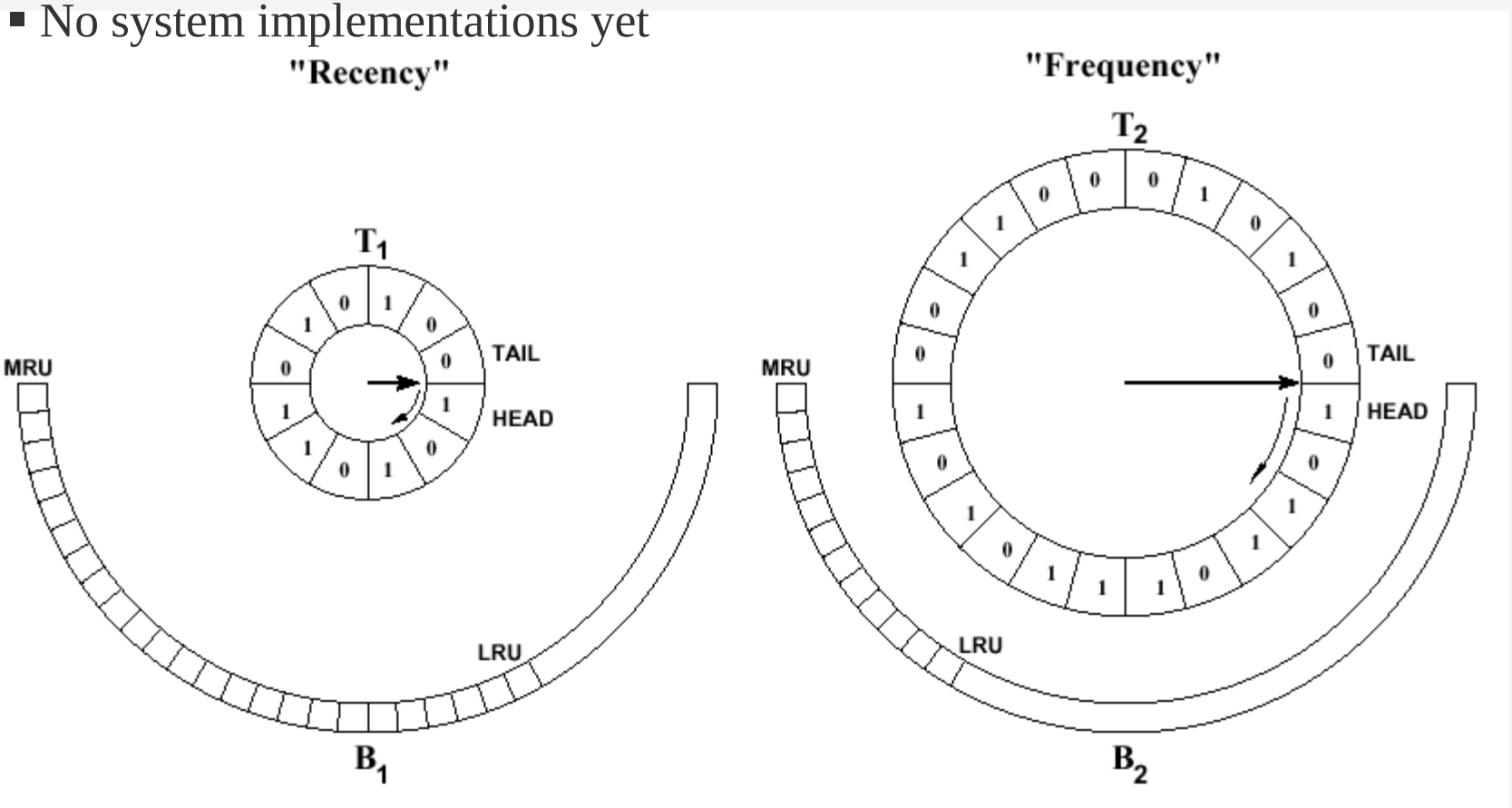
CAR: CLOCK with Adaptive Replacement

- Two clocks T1 and T2, one is for **cold pages** touched only once recently (Recency), another is for **hot pages** touched at least twice ("Frequency");
- Queues B1 and B2 are for pages recently **replaced** from T1 and T2;
- The memory allocations for T1 or T2 depend on **the ratio of references to B2 and B1**.



Limits of CAR

- A page that is **regularly accessed** with its reuse distance a little bit larger memory size has **no hits** in T1 or T2. (inherited LRU problem).
- A page in T2 can **stay in memory without any accesses** because frequency does not reflect ``reuse distance".
- No system implementations yet



Basic Ideas of CLOCK-Pro

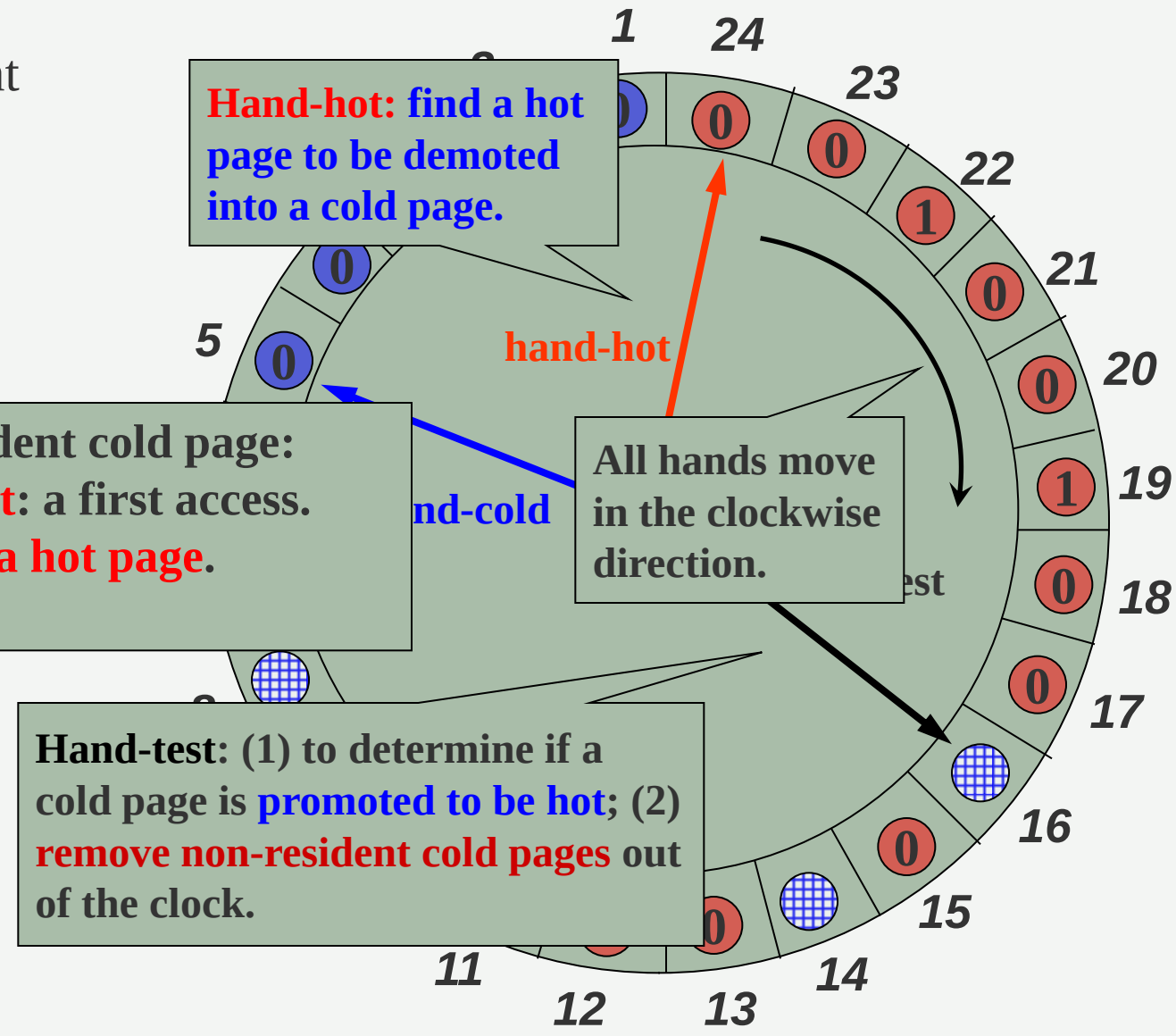
- It is an approximation of LIRS based on the CLOCK infrastructure.
- Pages categorized into two groups: **cold pages** **and** **hot pages** based on their reuse distances (or IRR).
- There are three hands: **Hand-hot** for hot pages, **Hand-cold** for cold pages, and **Hand-test** for running a reuse distance test for a block;
- The allocation of memory pages between hot pages (**M**hot) and cold pages (**M**cold) are **adaptively adjusted**. (**M** = **M**hot + **M**cold)
- All hot pages are resident (=Lir blocks), some cold pages are also resident (= Hir Blocks); keep track of **recently replaced pages** (=non-resident Hir blocks)

CLOCK-Pro Structure

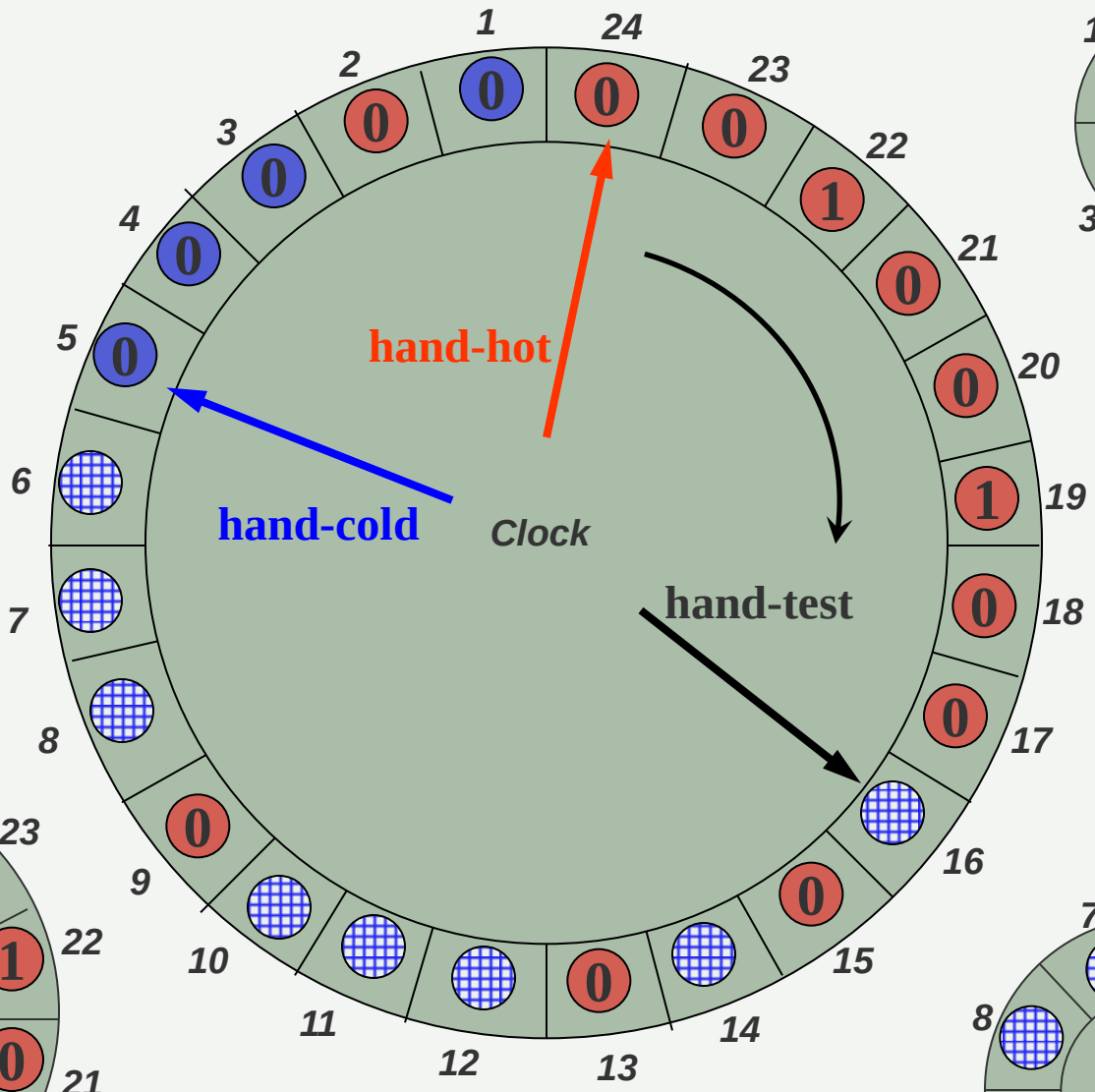
Cold resident

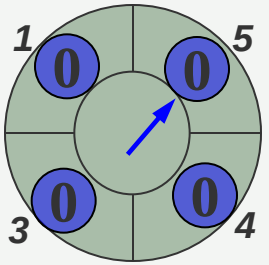
Hot

Cold non-resident

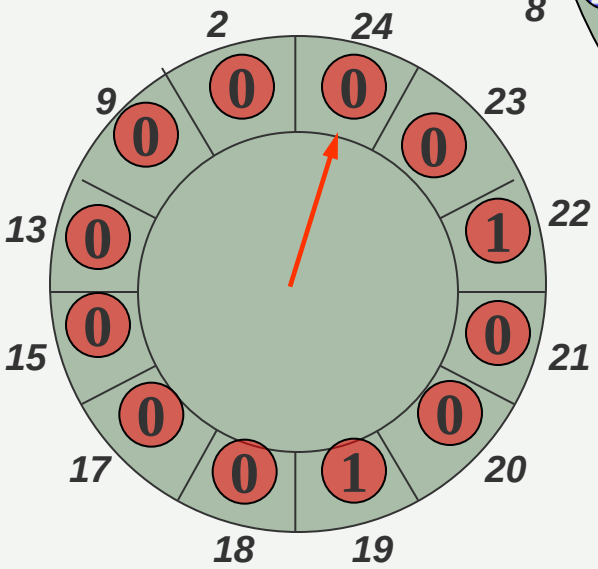


Accessing Sequence:
….27, 7, 26, 25, 4, 23

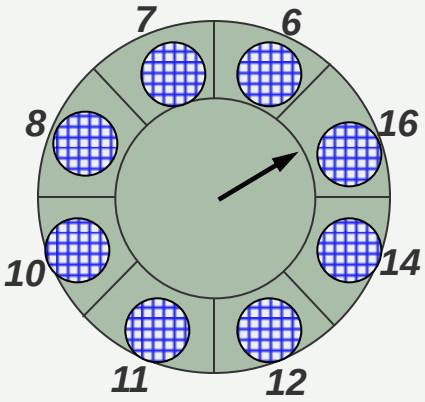




Resident
Cold Pages



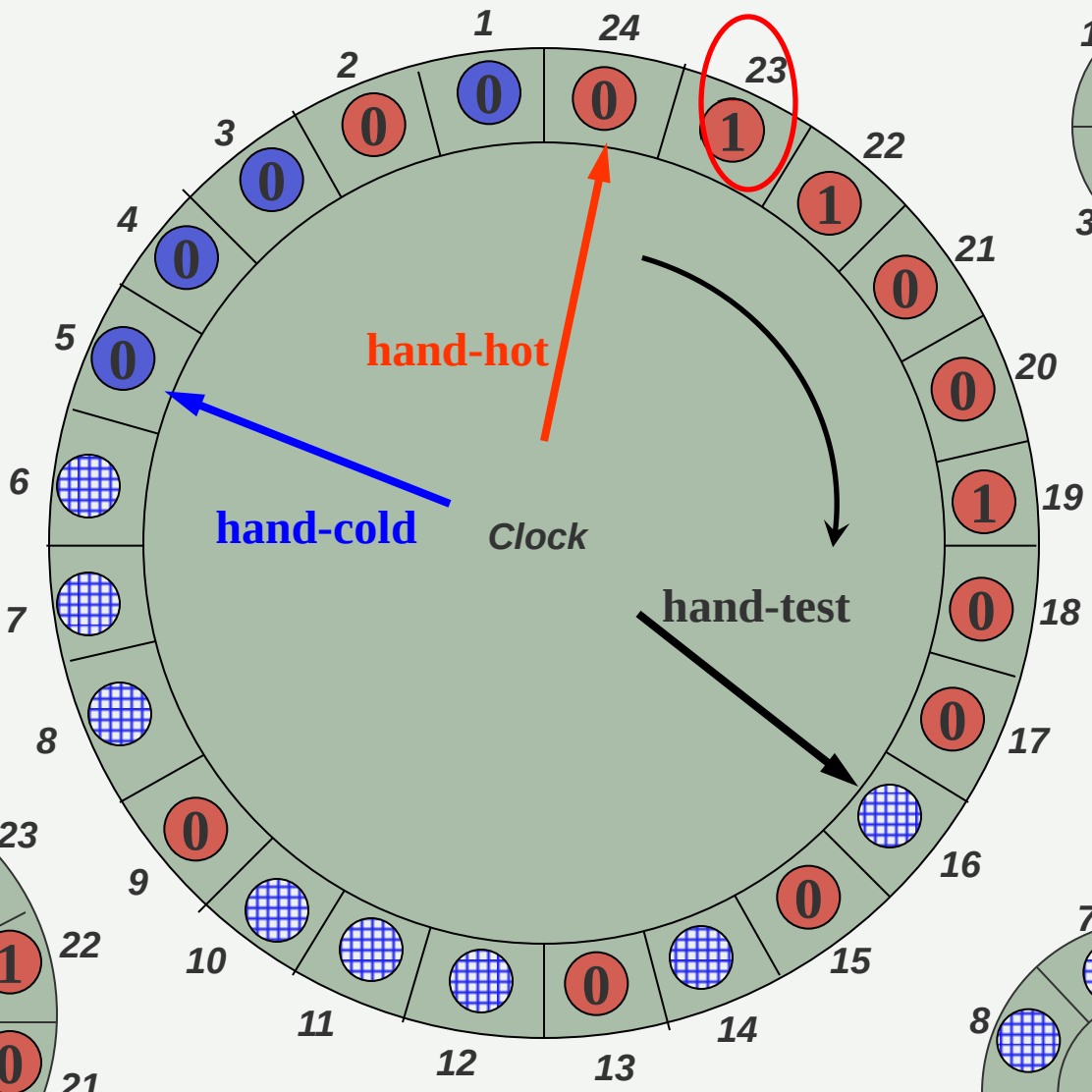
Hot Pages

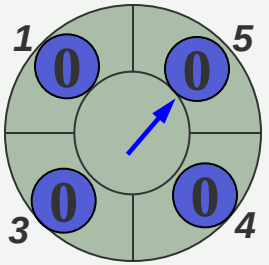


Non-resident Cold Pages

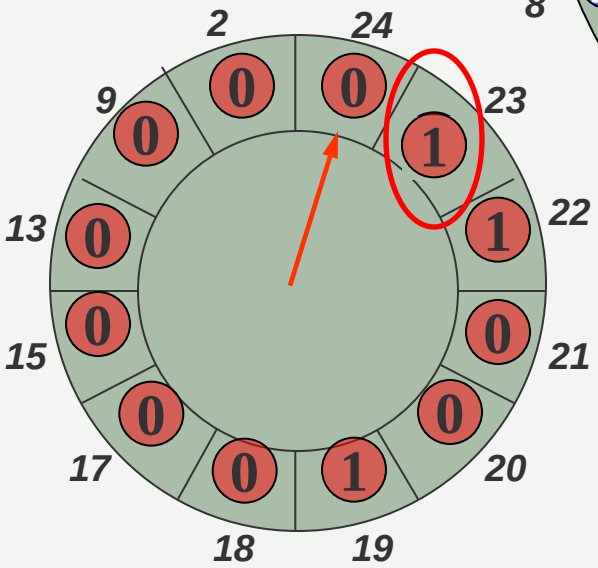
Current accessing:
Page 23

Hit!
Set reference bit of
Page 23 to 1.
No other operation

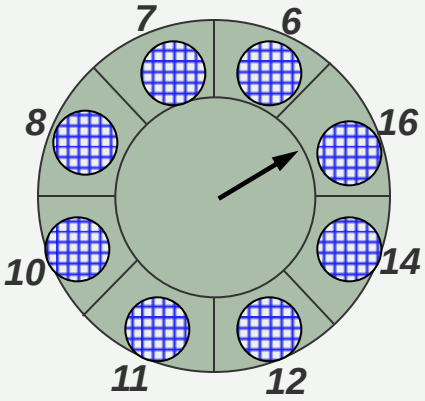




Resident
Cold Pages



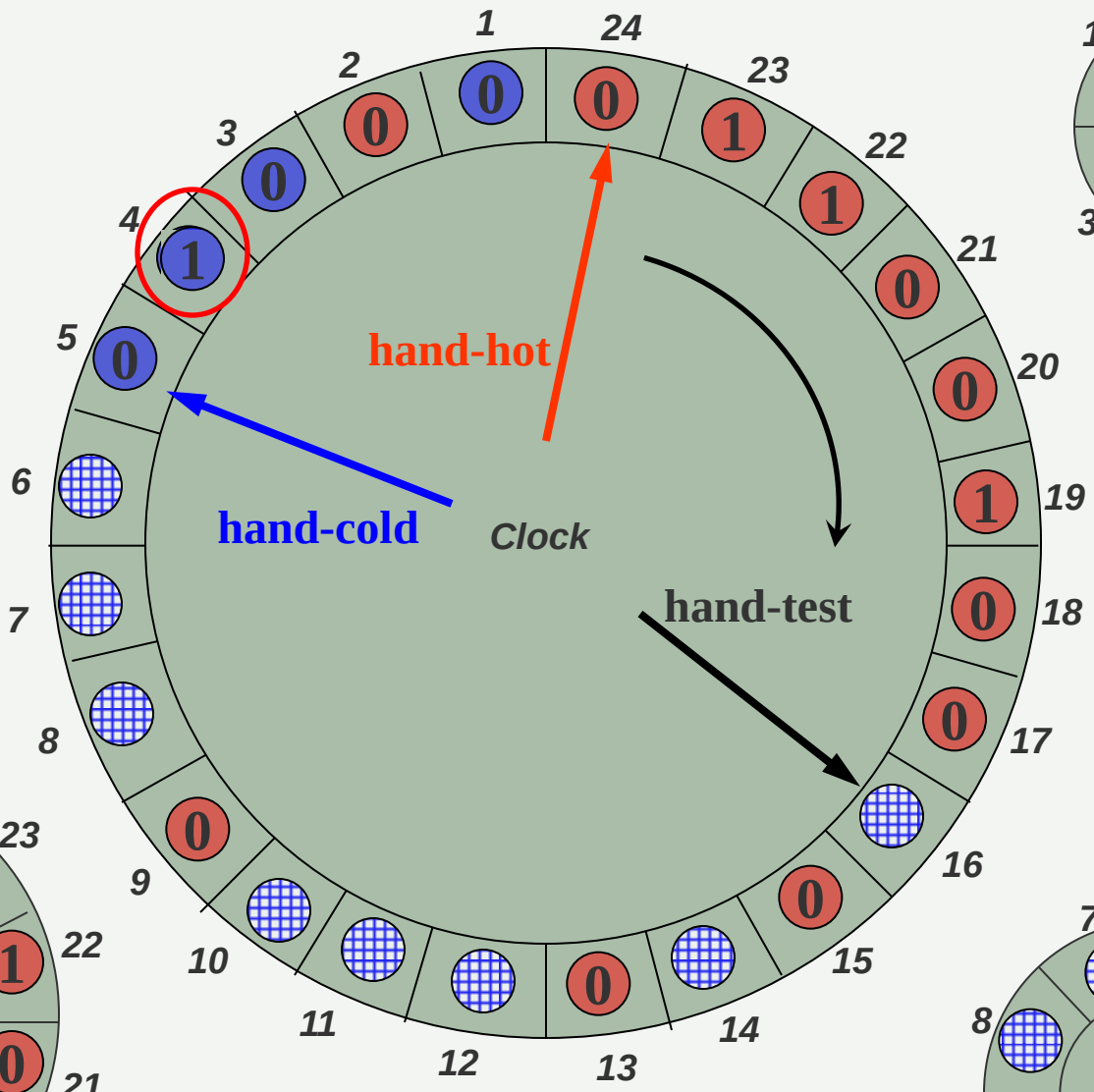
Hot Pages

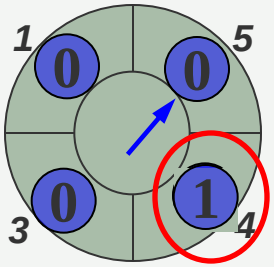


Non-resident Cold Pages

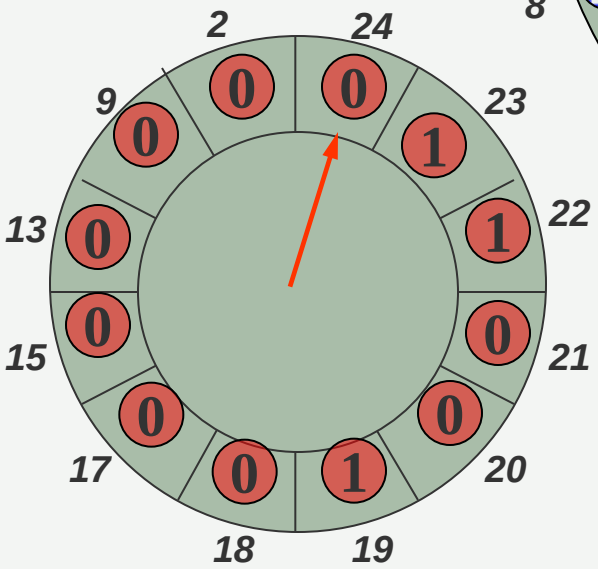
Current accessing:
Page 4

Hit!
Set reference bit of
Page 4 to 1.
No other operation

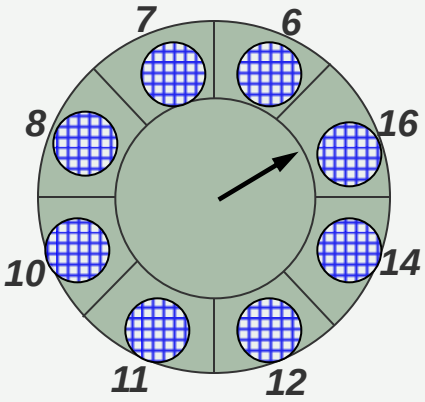




Resident
Cold Pages



Hot Pages



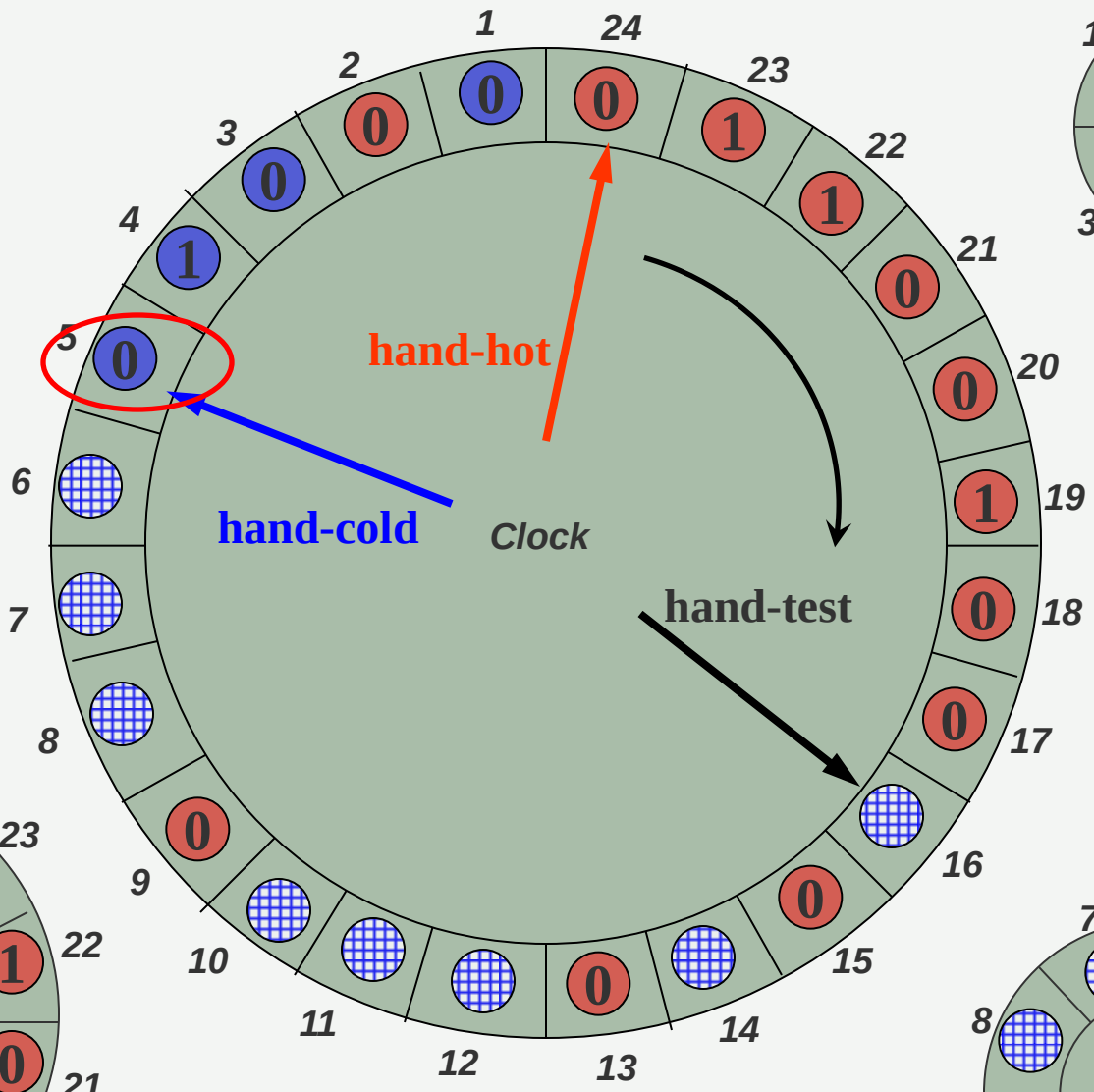
Non-resident Cold Pages

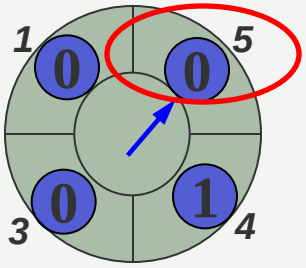
Current accessing: Page 25

Miss!

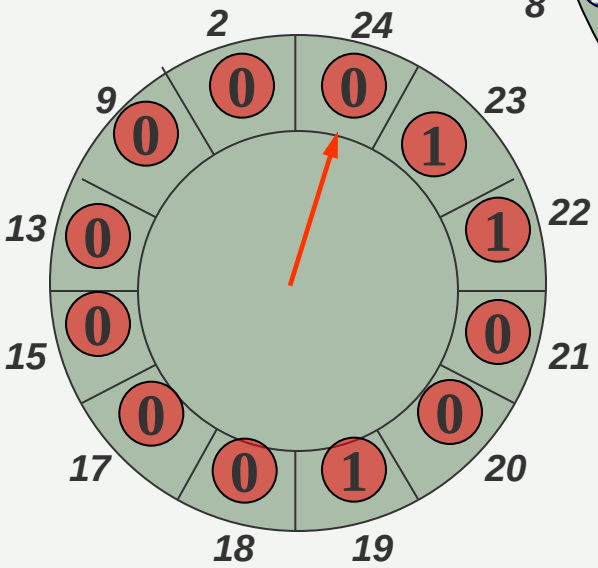
(1) run hand-cold--- reclaim the first met cold page with ref bit 0 (Page 5)

(2) Reclaim cold page 5 and remove it from resident cold page list

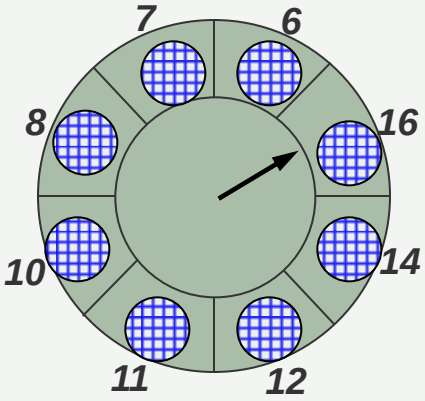




Resident Cold Pages



Hot Pages



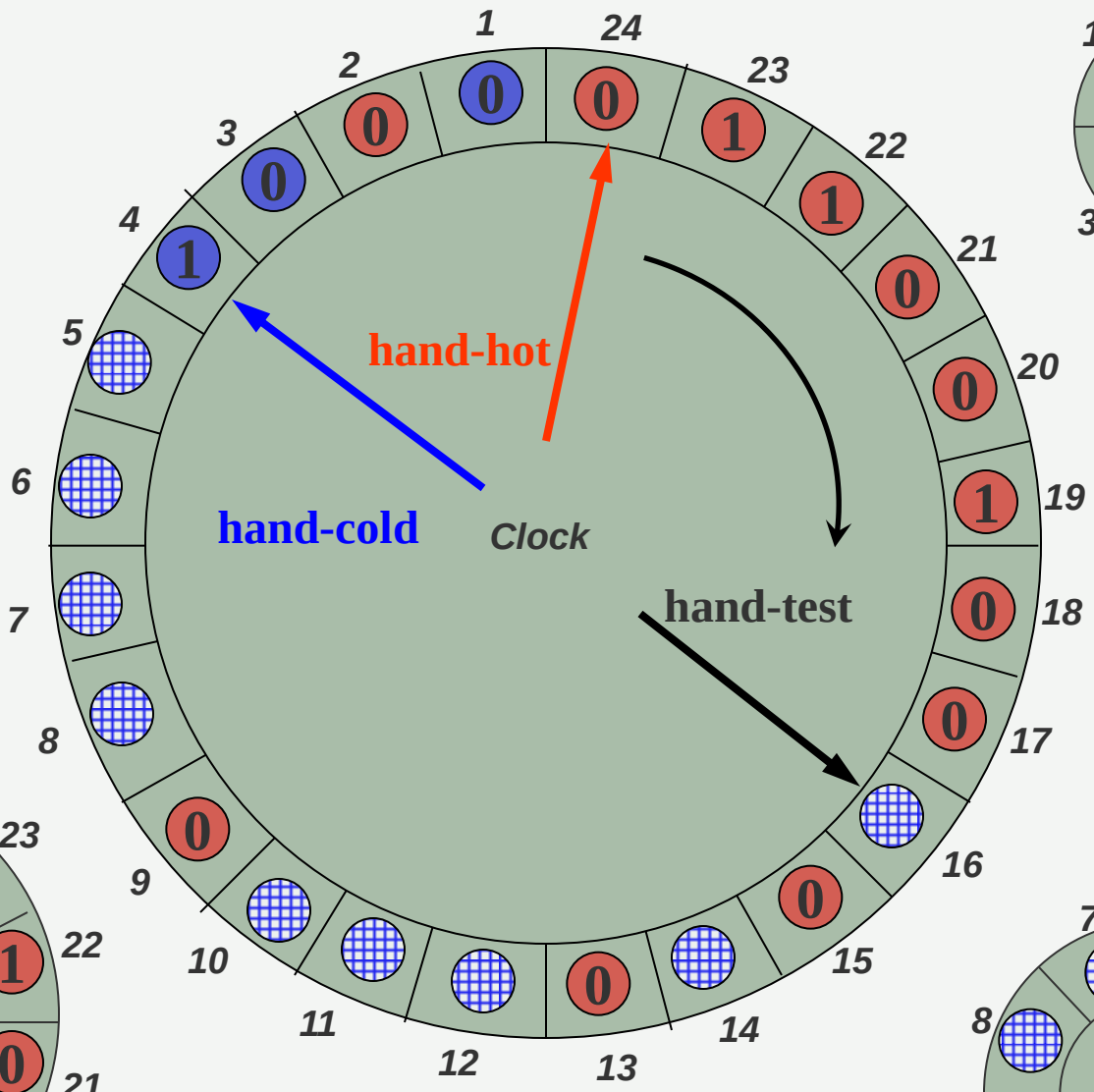
Non-resident Cold Pages

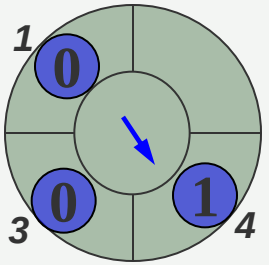
Current accessing: Page 25

(3) Leave non-resident cold page 5 in the original position of clock list

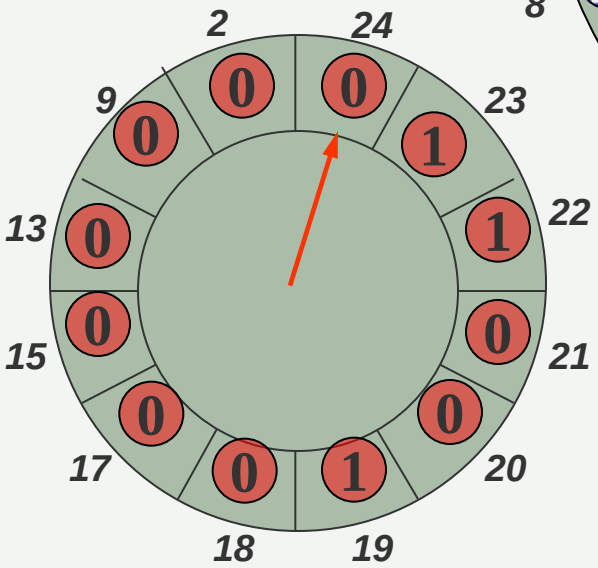
(4) Add page 5 into non-resident cold page list

(5) run hand-test to find a position for page 5

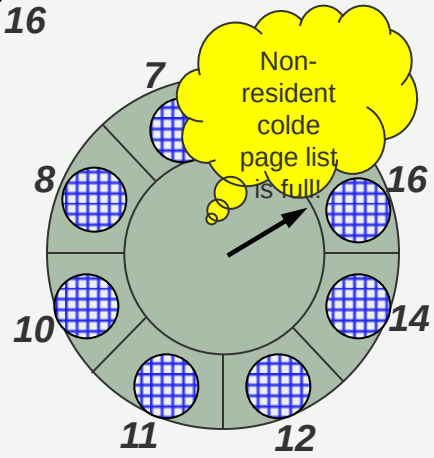




Resident Cold Pages



Hot Pages

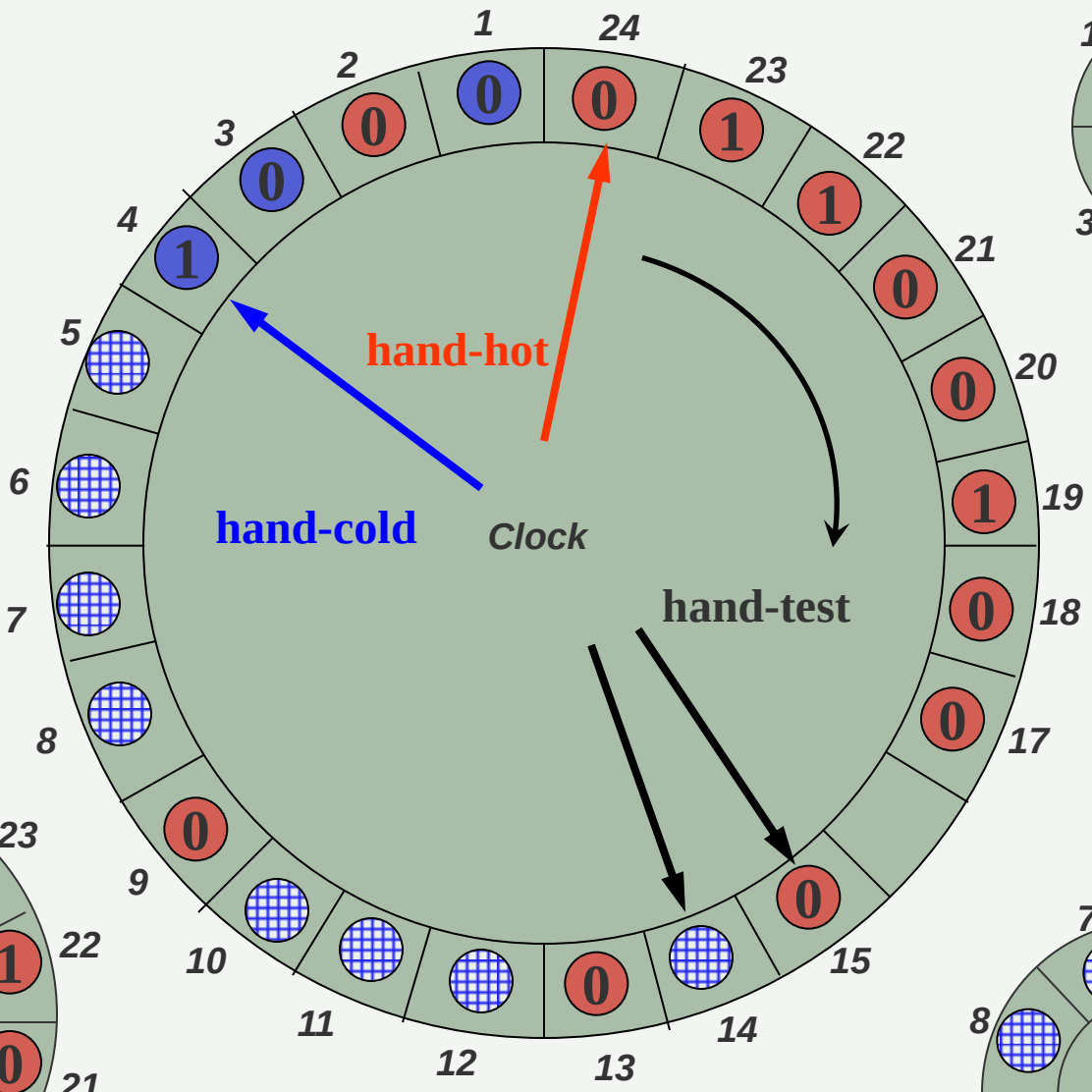


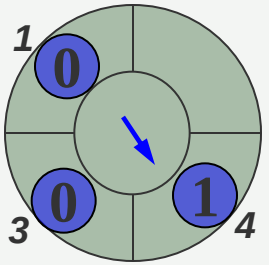
Non-resident Cold Pages

Current accessing:
Page 25

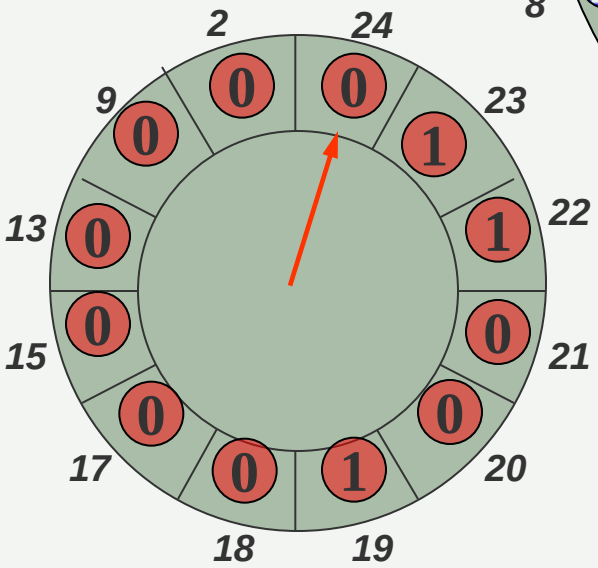
(5) Add page 5 to non-resident cold page list

Reorganize clock list

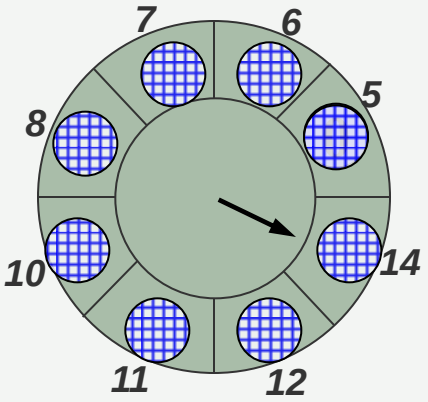




Resident
Cold Pages



Hot Pages



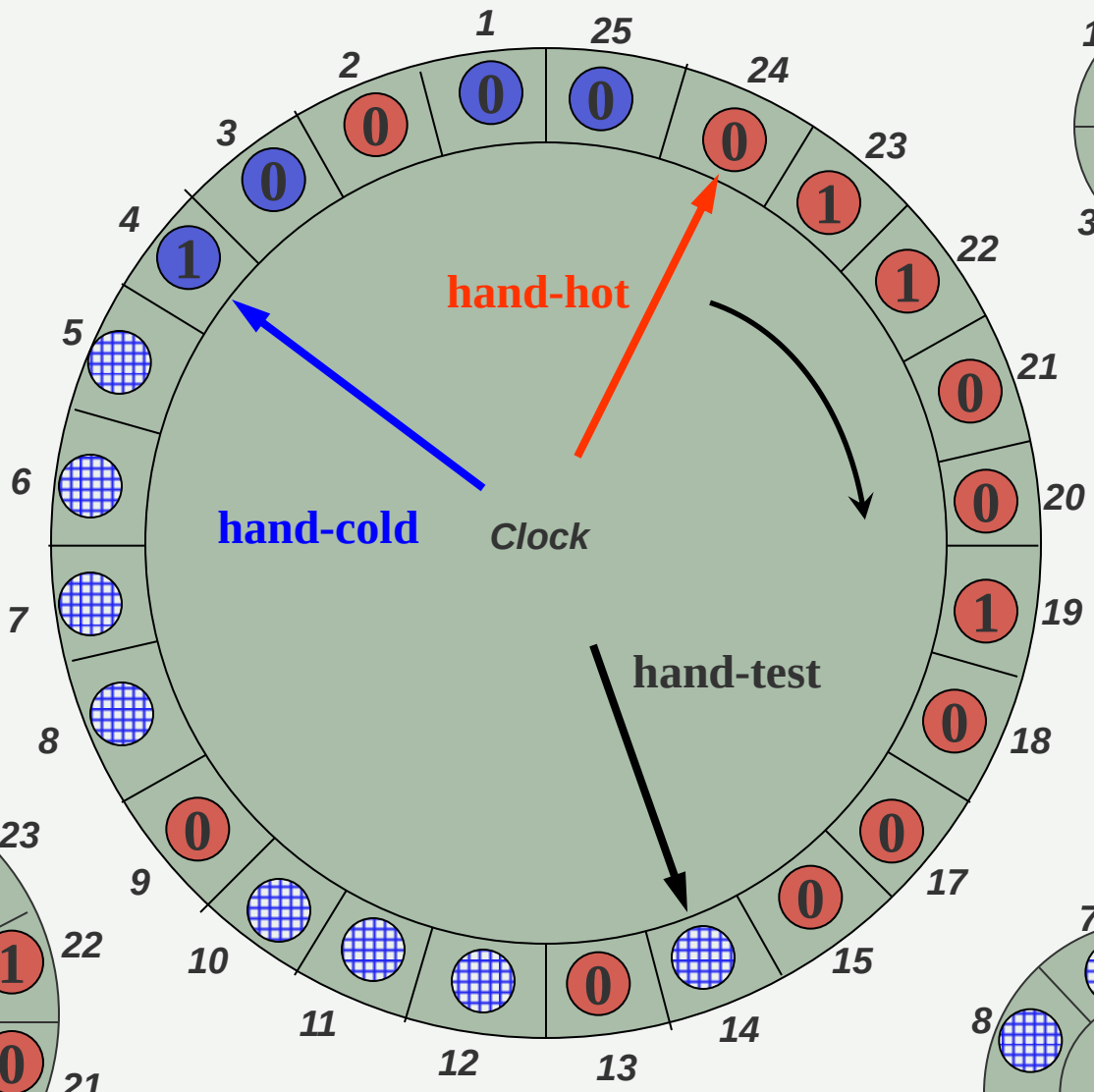
Non-resident Cold Pages

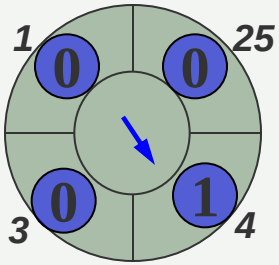
Current accessing: Page 25

Now we already have one empty room for page 25

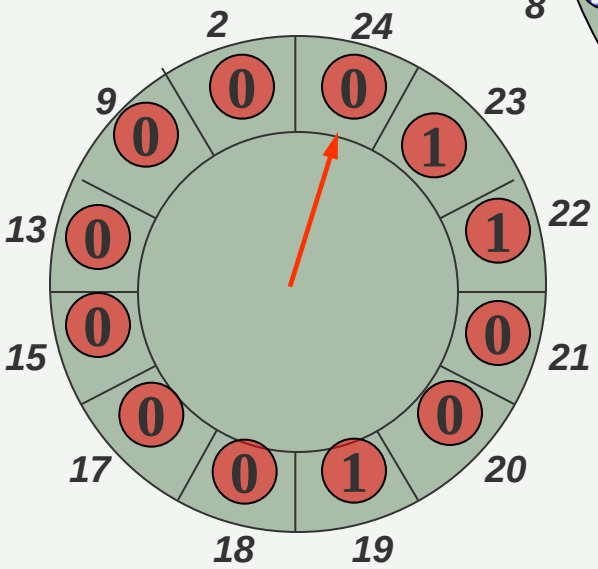
(6) Check page 25 in non-resident page list---it is not in it

(7) Add page 25 into clock and resident cold page list

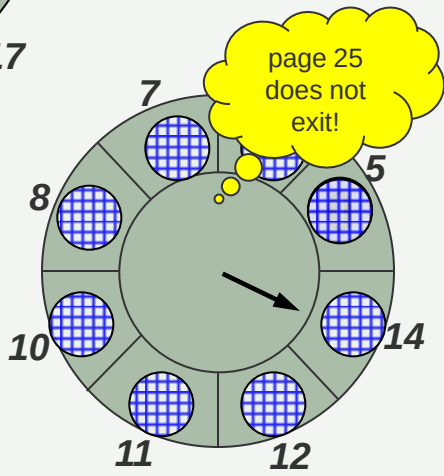




Resident Cold Pages



Hot Pages

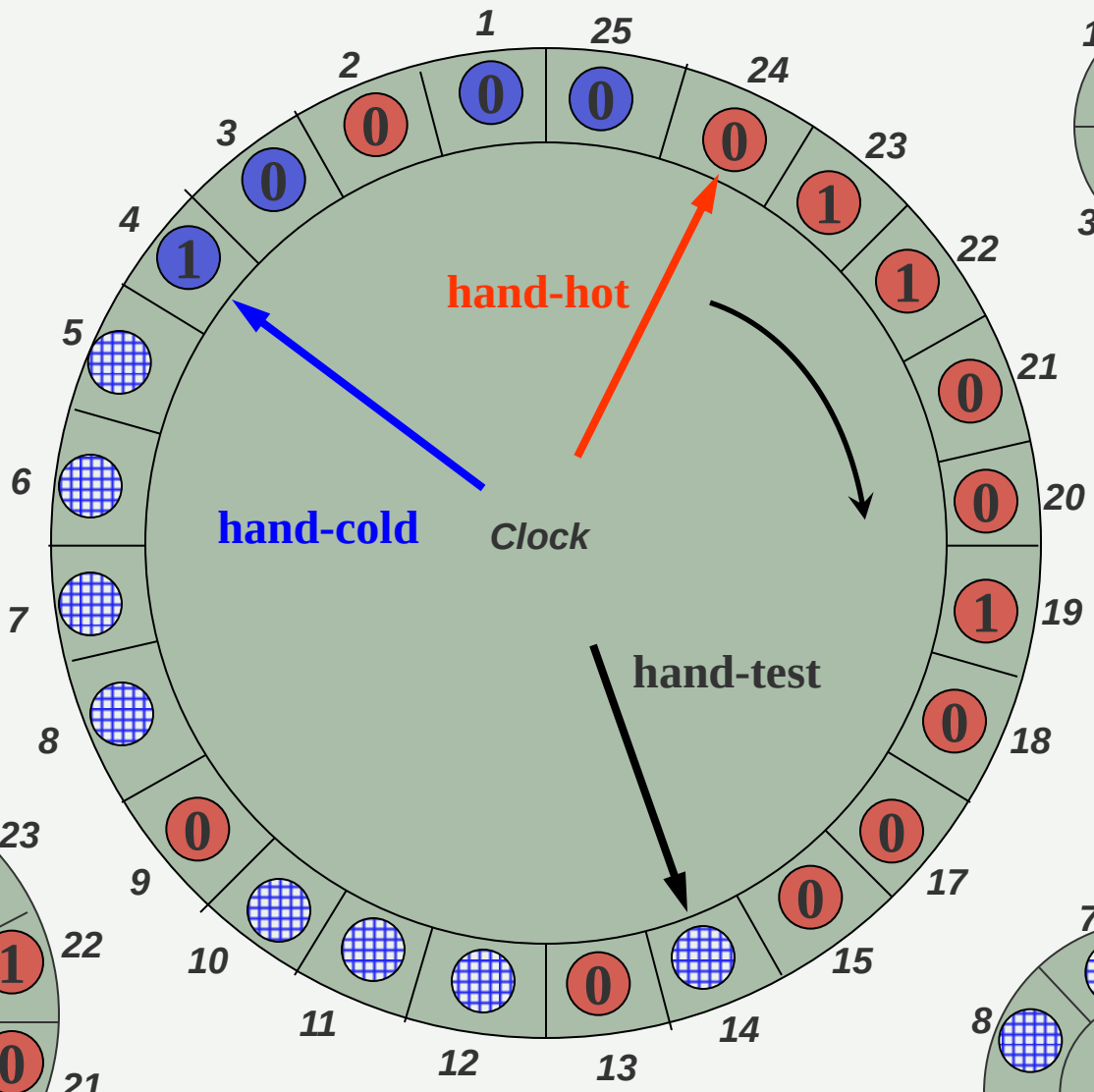


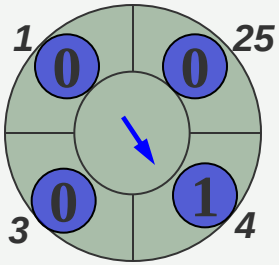
Non-resident Cold Pages

Current accessing: Page 26

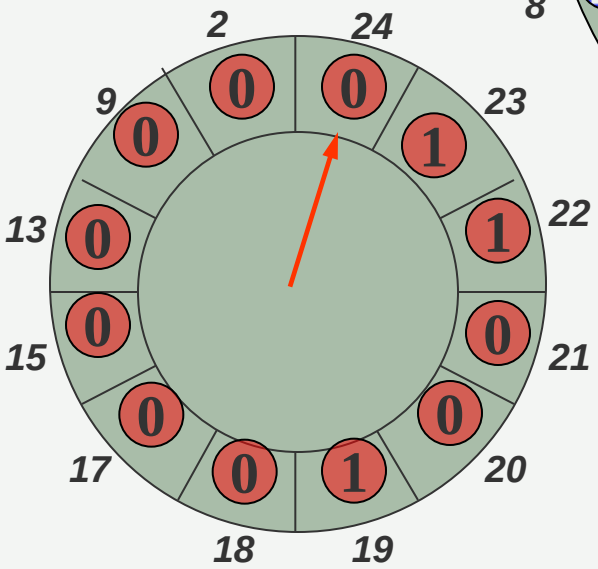
Miss!

(1) run hand-cold to reach the first met cold resident page with reference bit 0

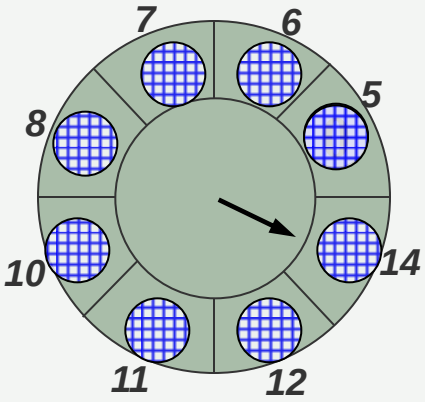




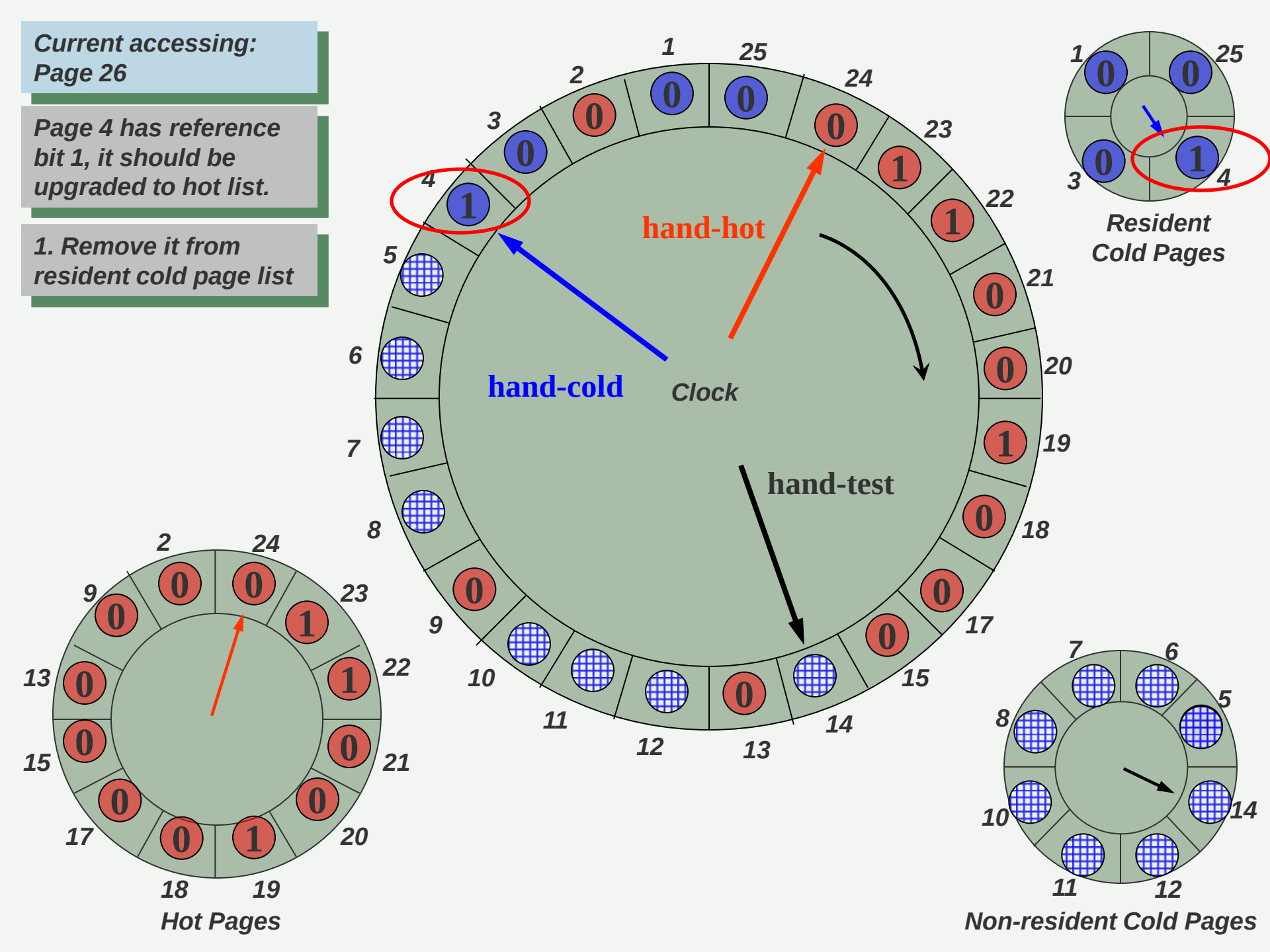
Resident Cold Pages

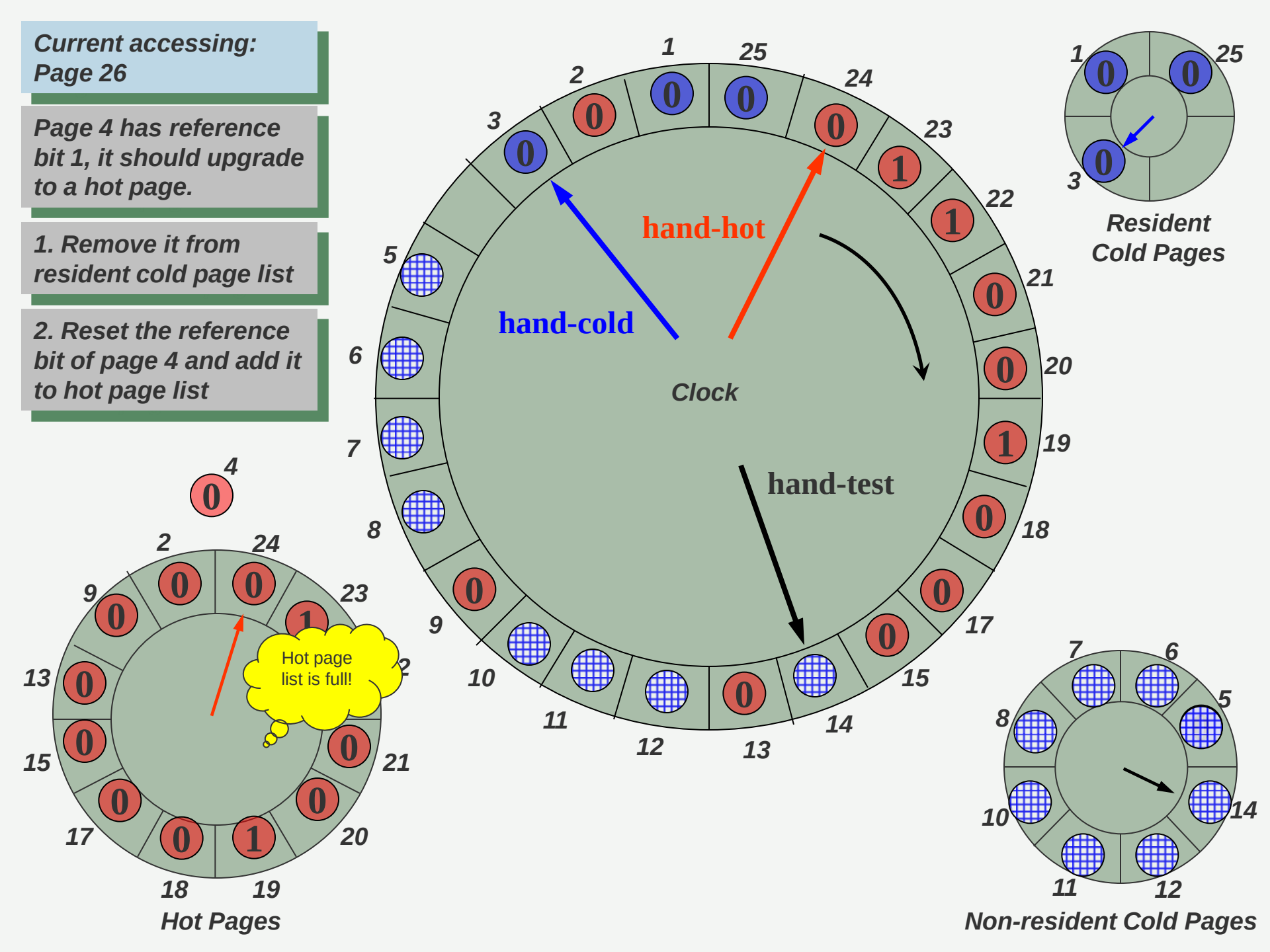


Hot Pages



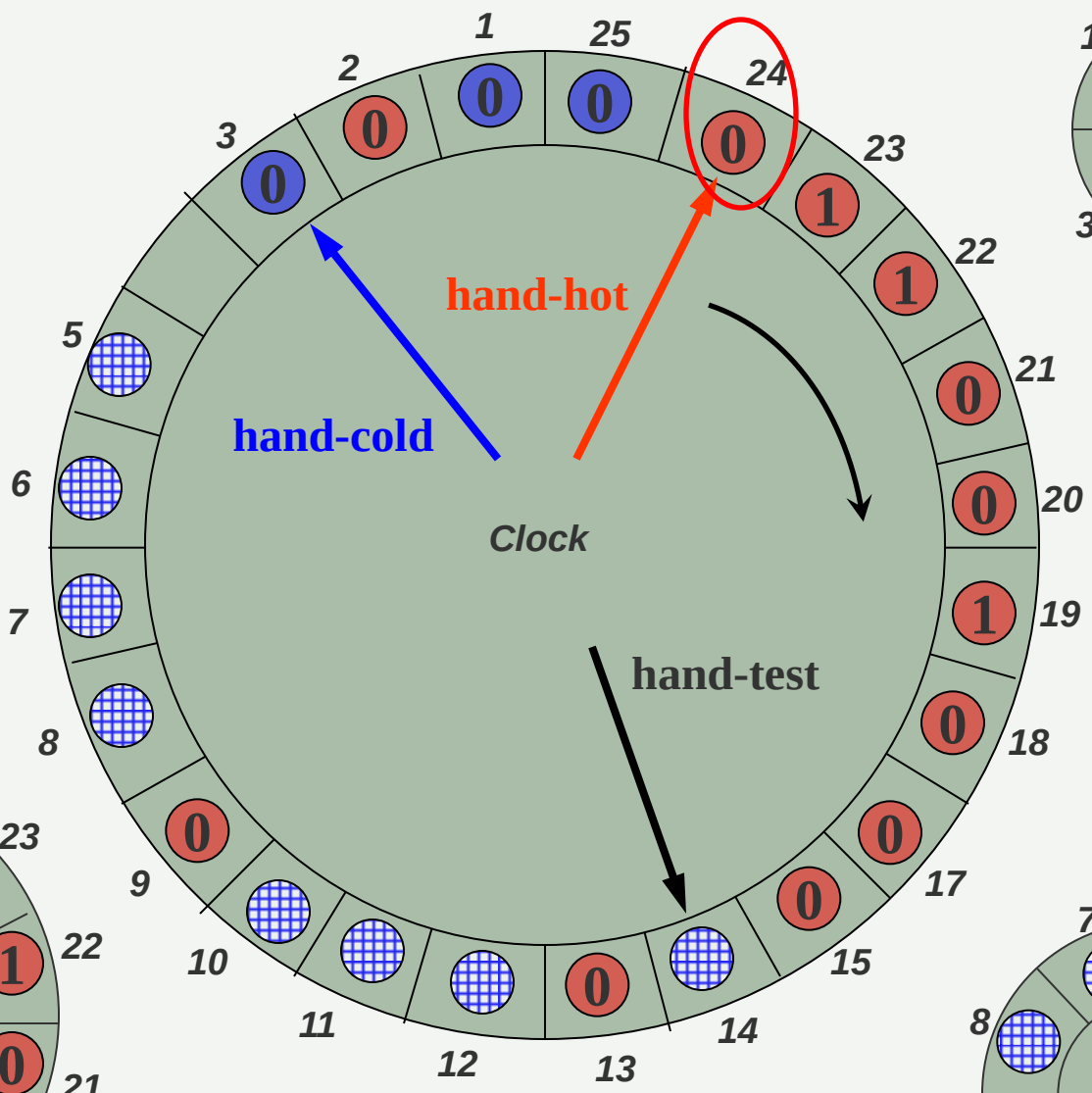
Non-resident Cold Pages

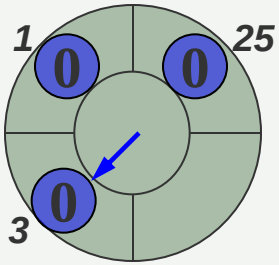

Current accessing:
Page 26
Page 4 has reference bit 1, it should be upgraded to hot list.
1. Remove it from resident cold page list
hand-hot
hand-cold
Clock
hand-test
Resident Cold Pages
Hot Pages
Non-resident Cold Pages


Current accessing:
Page 26
Page 4 has reference bit 1, it should upgrade to a hot page.
1. Remove it from resident cold page list
2. Reset the reference bit of page 4 and add it to hot page list
hand-hot
hand-cold
hand-test
Clock
Hot page list is full!
Hot Pages
Resident Cold Pages
Non-resident Cold Pages

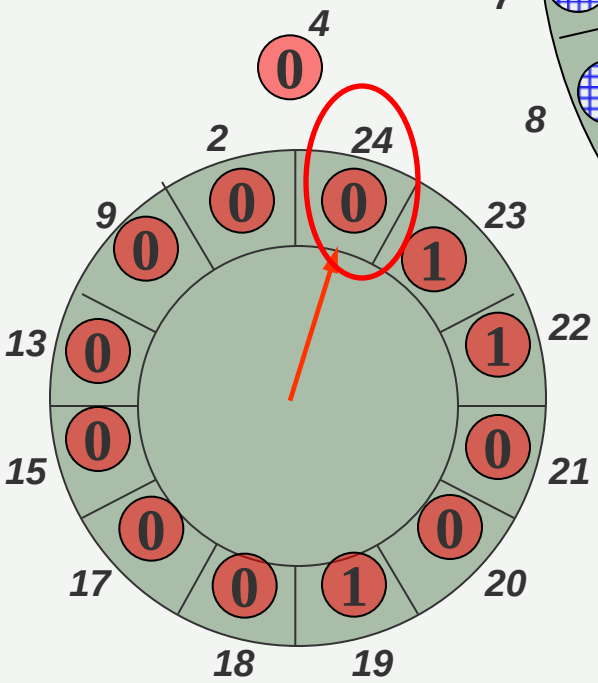
Current accessing: Page 26

***run hand-hot**: degrade the first met hot page with reference bit 0 to a cold resident page*

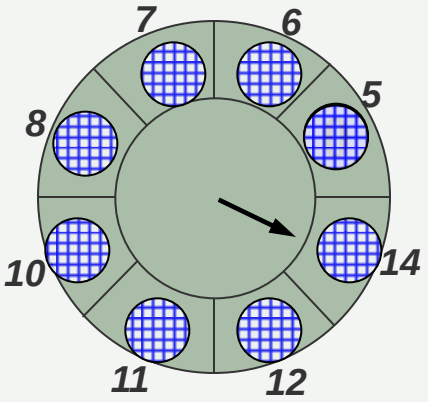




Resident Cold Pages



Hot Pages

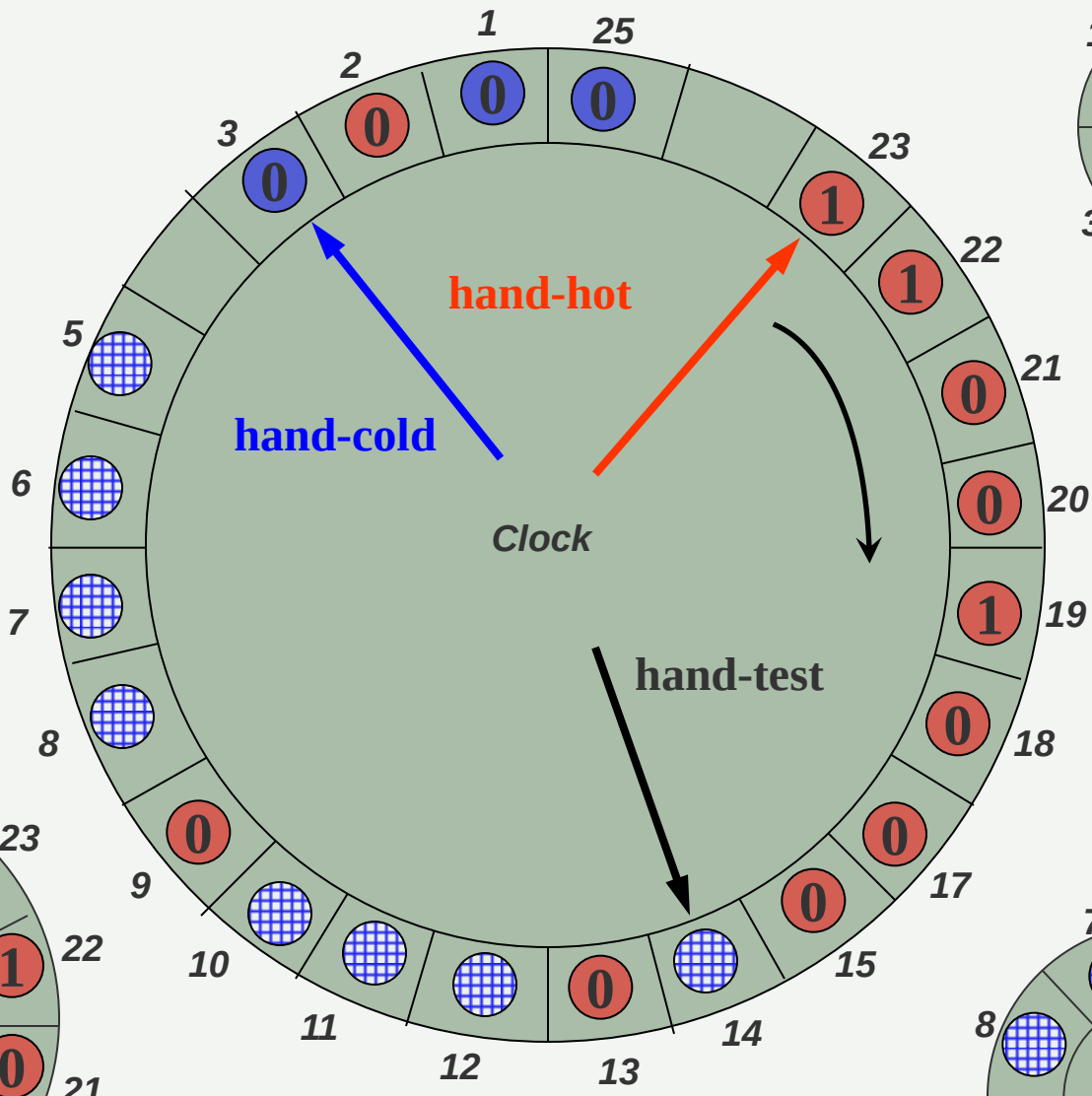


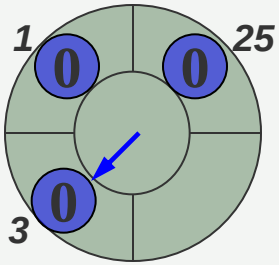
Non-resident Cold Pages

Current accessing: Page 26

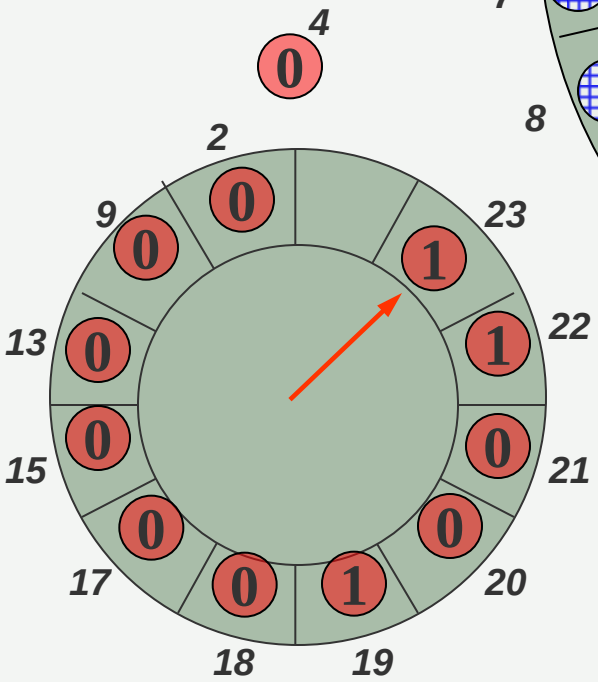
Run hand-hot: degrade first met hot page with ref bit 0 to a cold resident page

Reorganize clock list

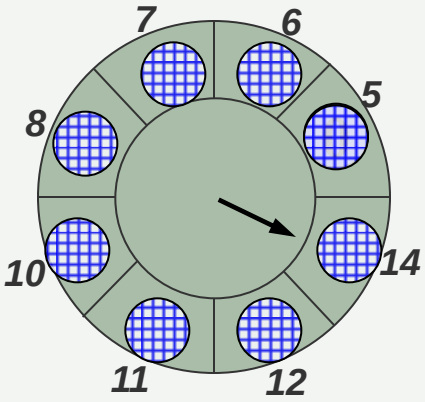




Resident Cold Pages



Hot Pages

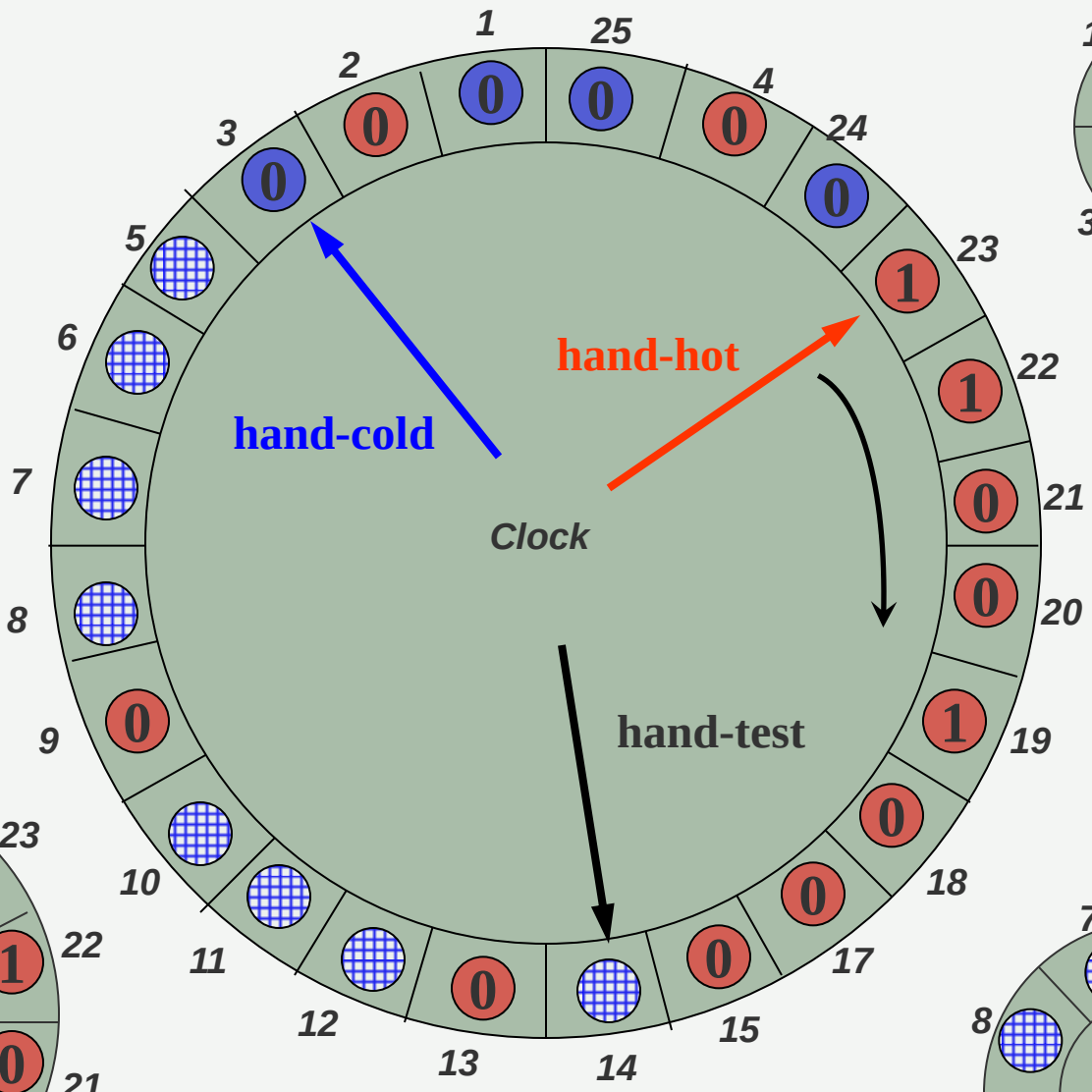


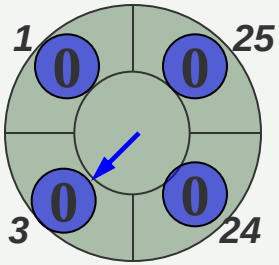
Non-resident Cold Pages

Current accessing: Page 26

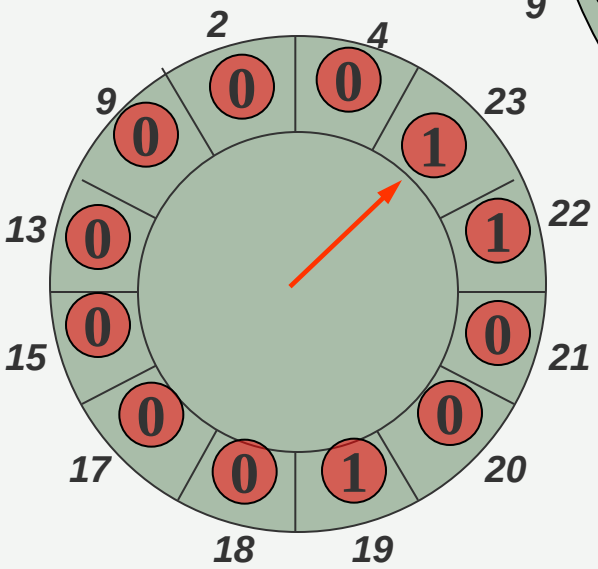
Page 4 is added into hot page list.

Page 24 is added into cold page list.

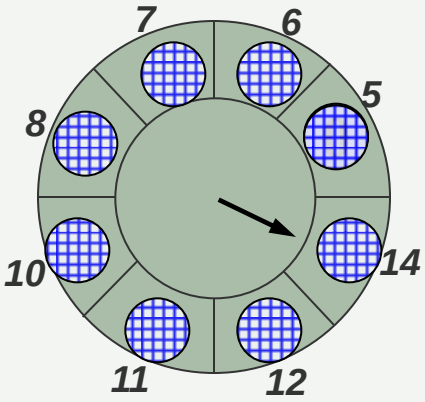




Resident Cold Pages



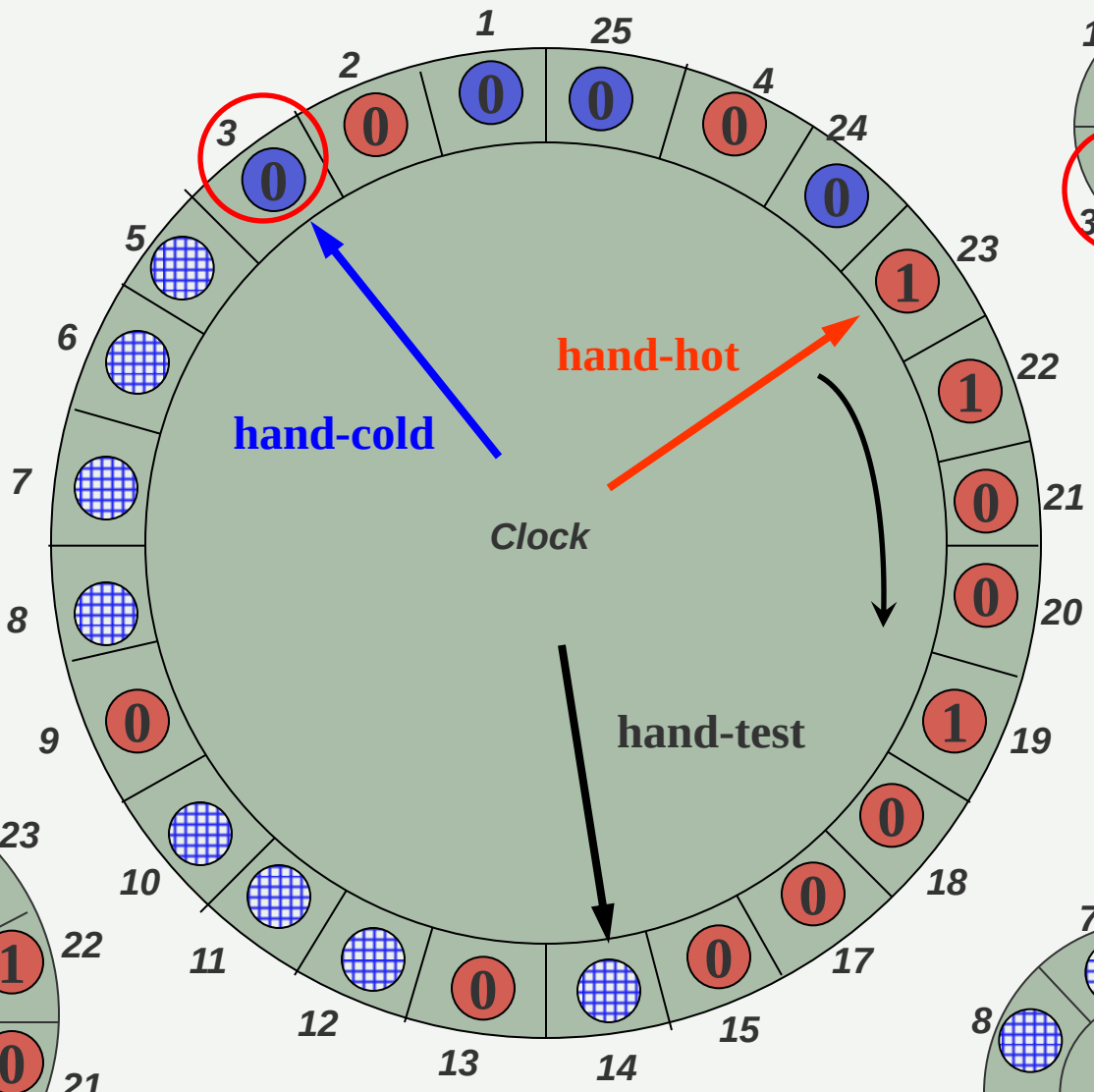
Hot Pages

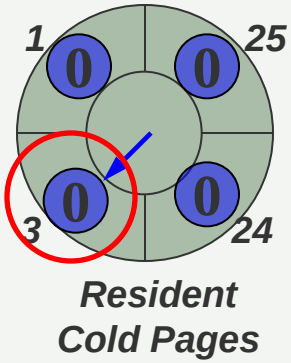


Non-resident Cold Pages

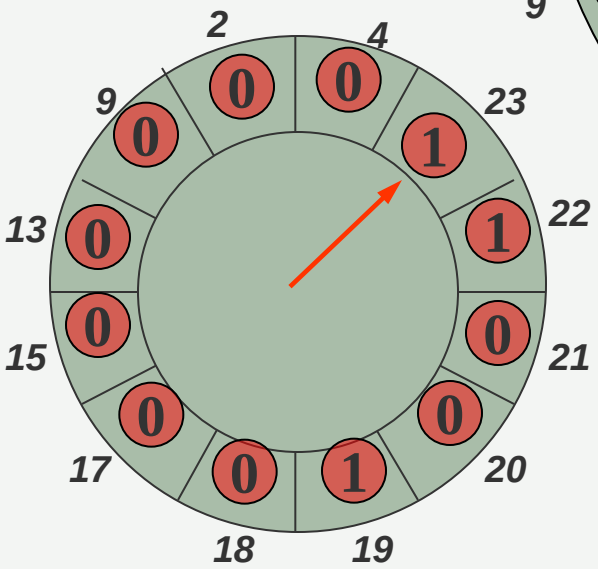
Current accessing:
Page 26

***run hand-cold**:*
Page 3 has reference bit 0, reclaim it and move hand-cold to the next position

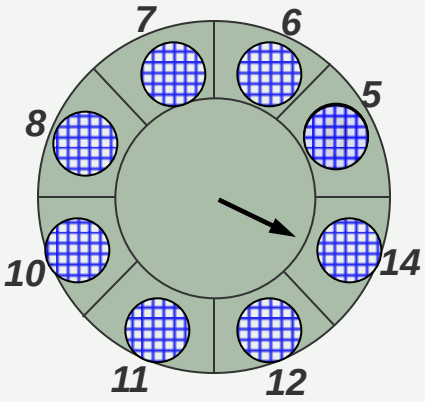




Resident Cold Pages



Hot Pages



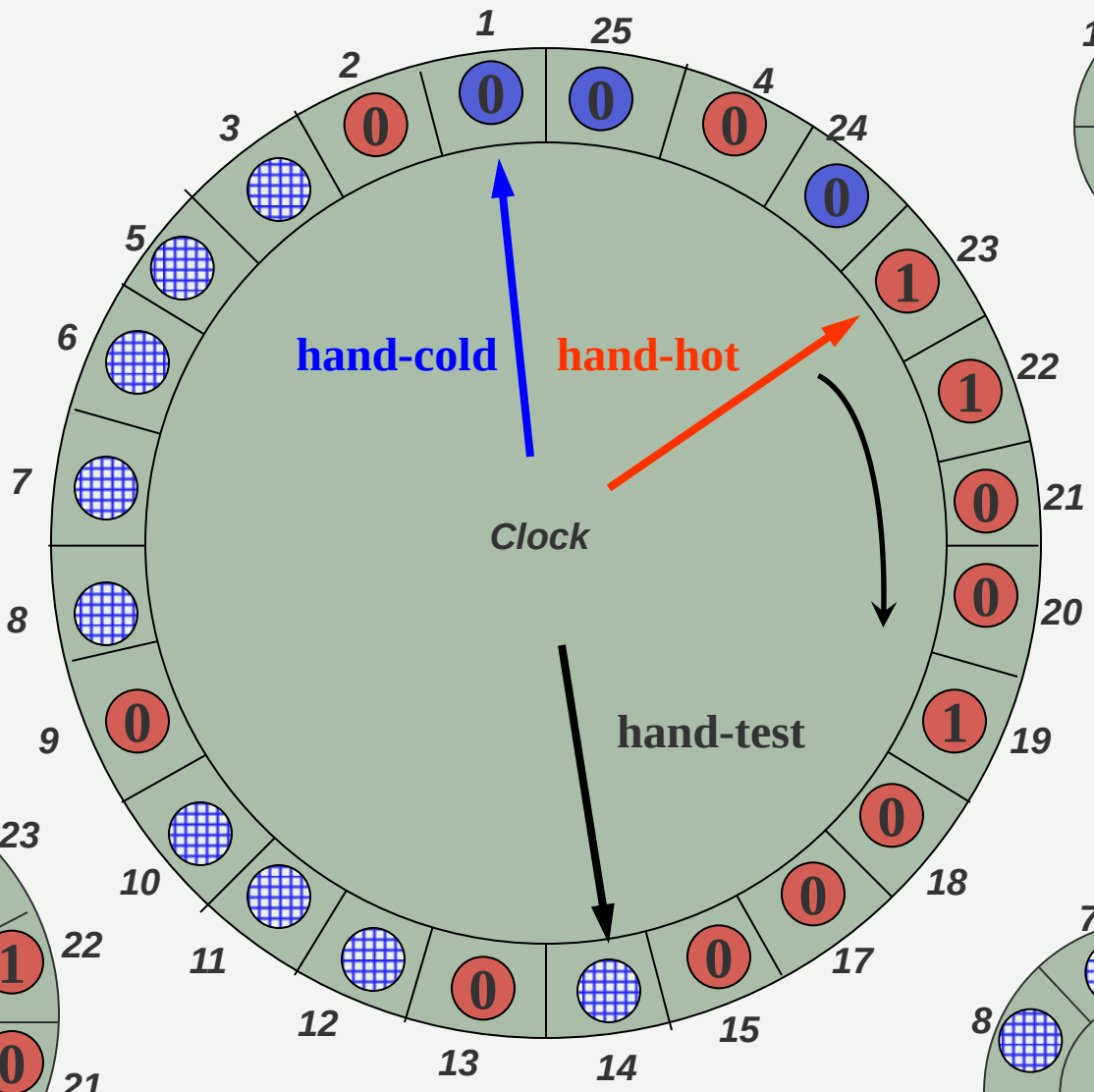
Non-resident Cold Pages

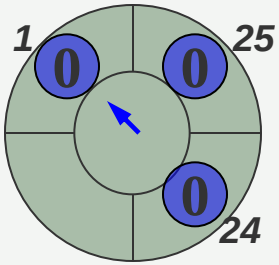
Current accessing: Page 26

Reclaim page 3 and add it into non-resident cold page list.

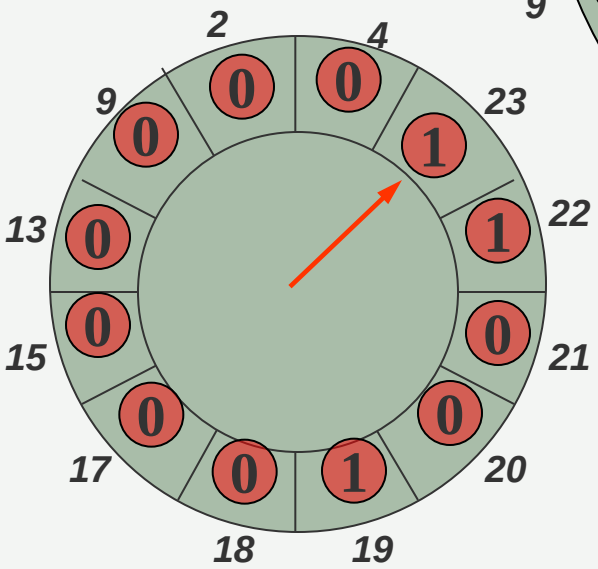
1. Leave non-resident cold page 3 in the old position in clock list.

2. run hand-test: move 14 from non-resident cold page list.

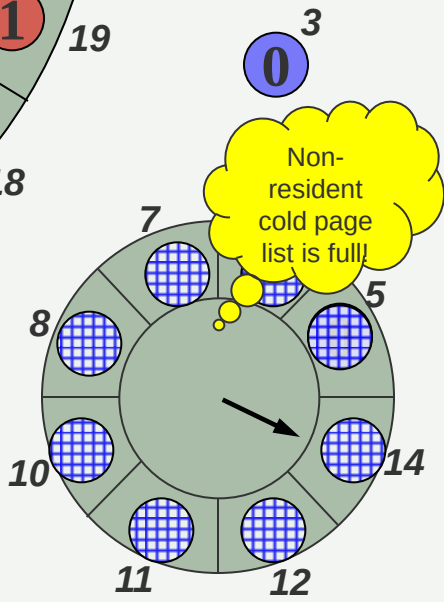




Resident Cold Pages



Hot Pages

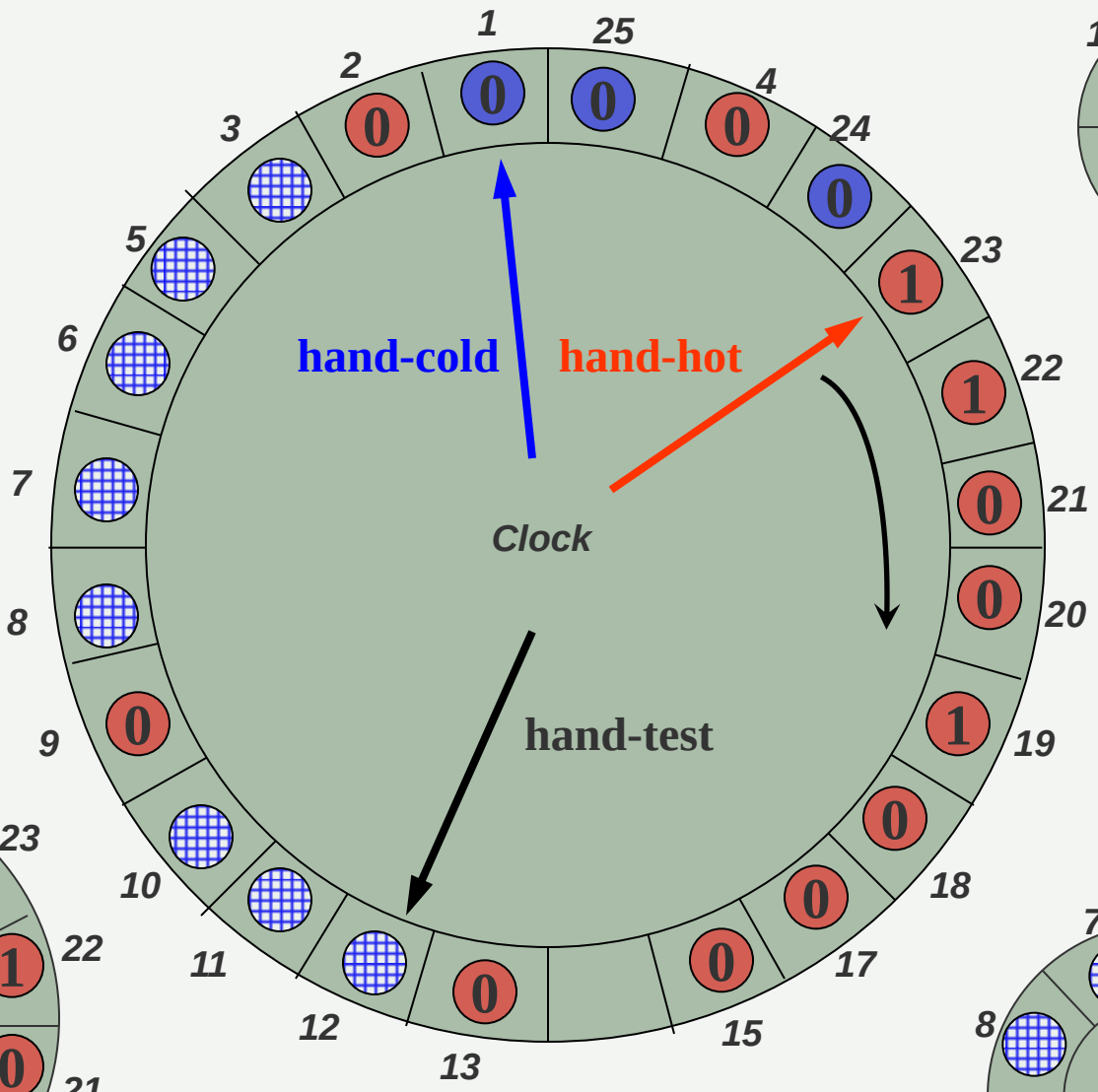


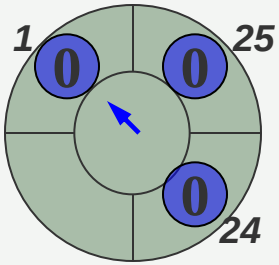
Non-resident Cold Pages

Current accessing: Page 26

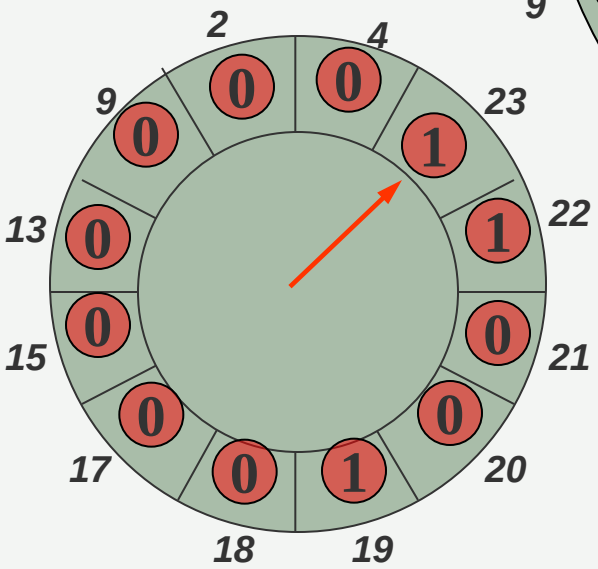
Add page 3 into non-resident cold page list.

Reorganize clock list

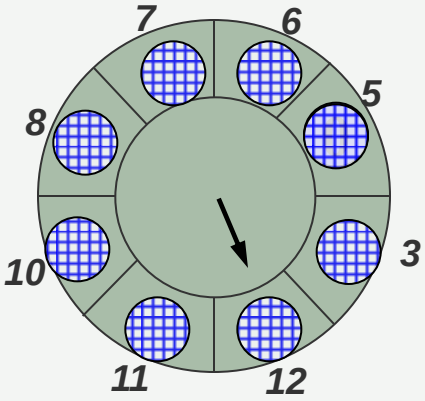




Resident Cold Pages



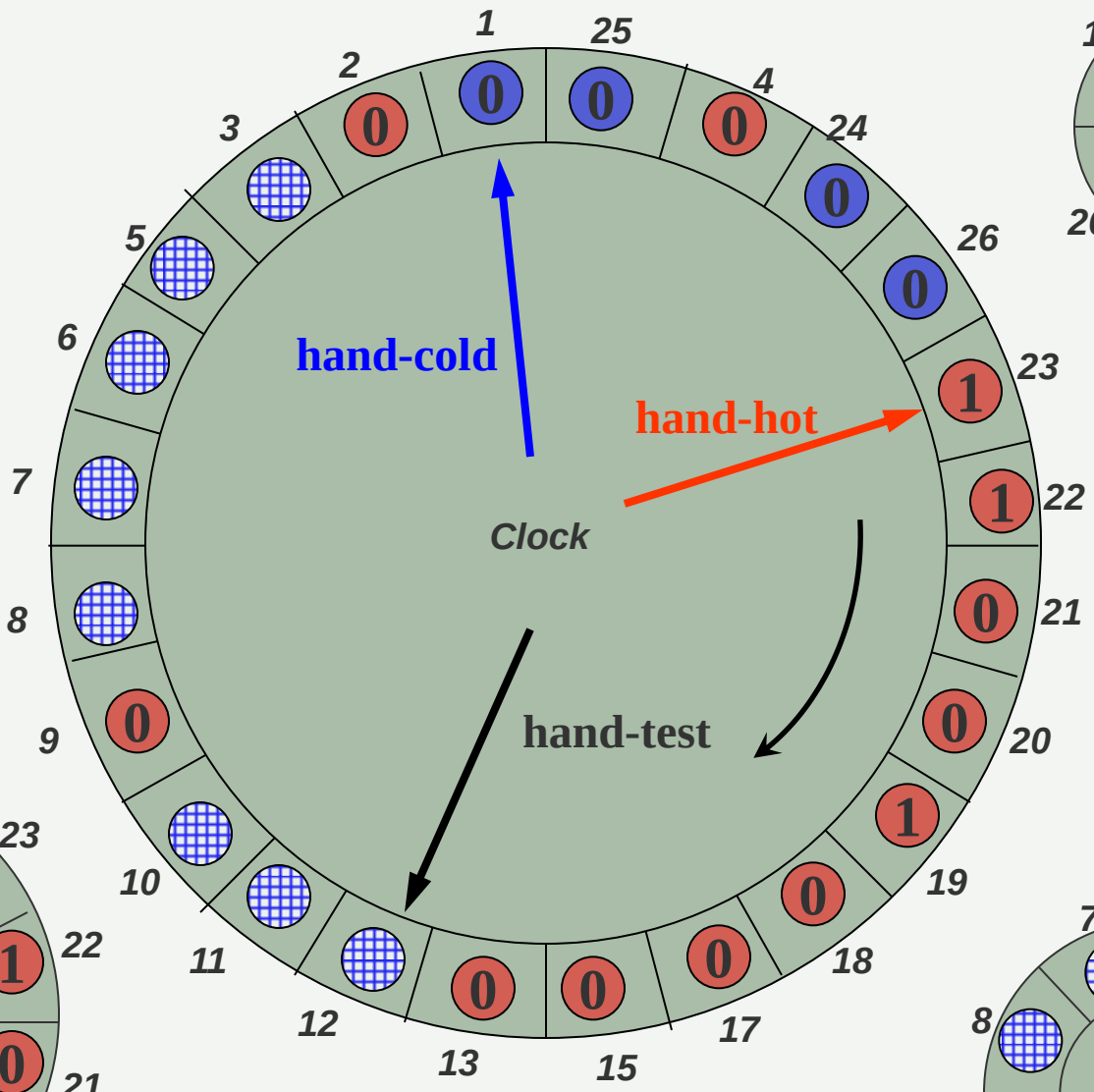
Hot Pages

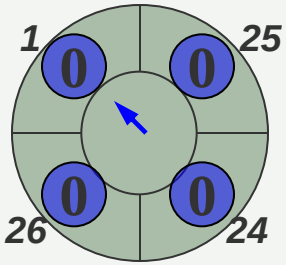


Non-resident Cold Pages

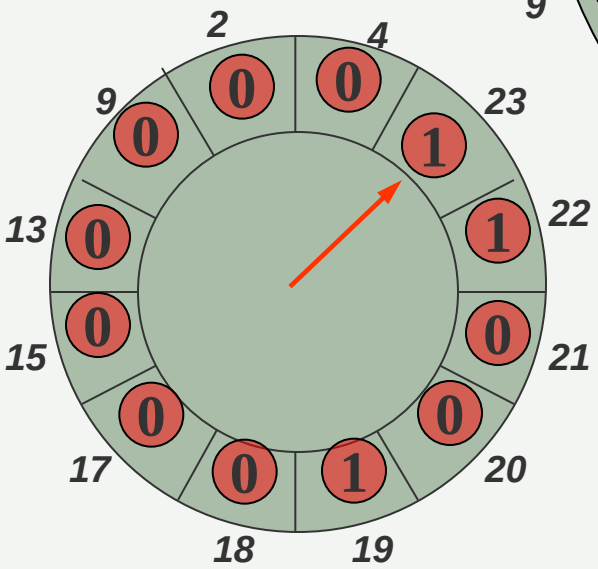
Current accessing: Page 26

Add page 26 into resident cold page list and clock list

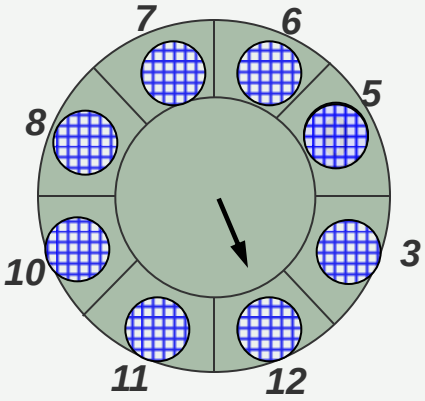




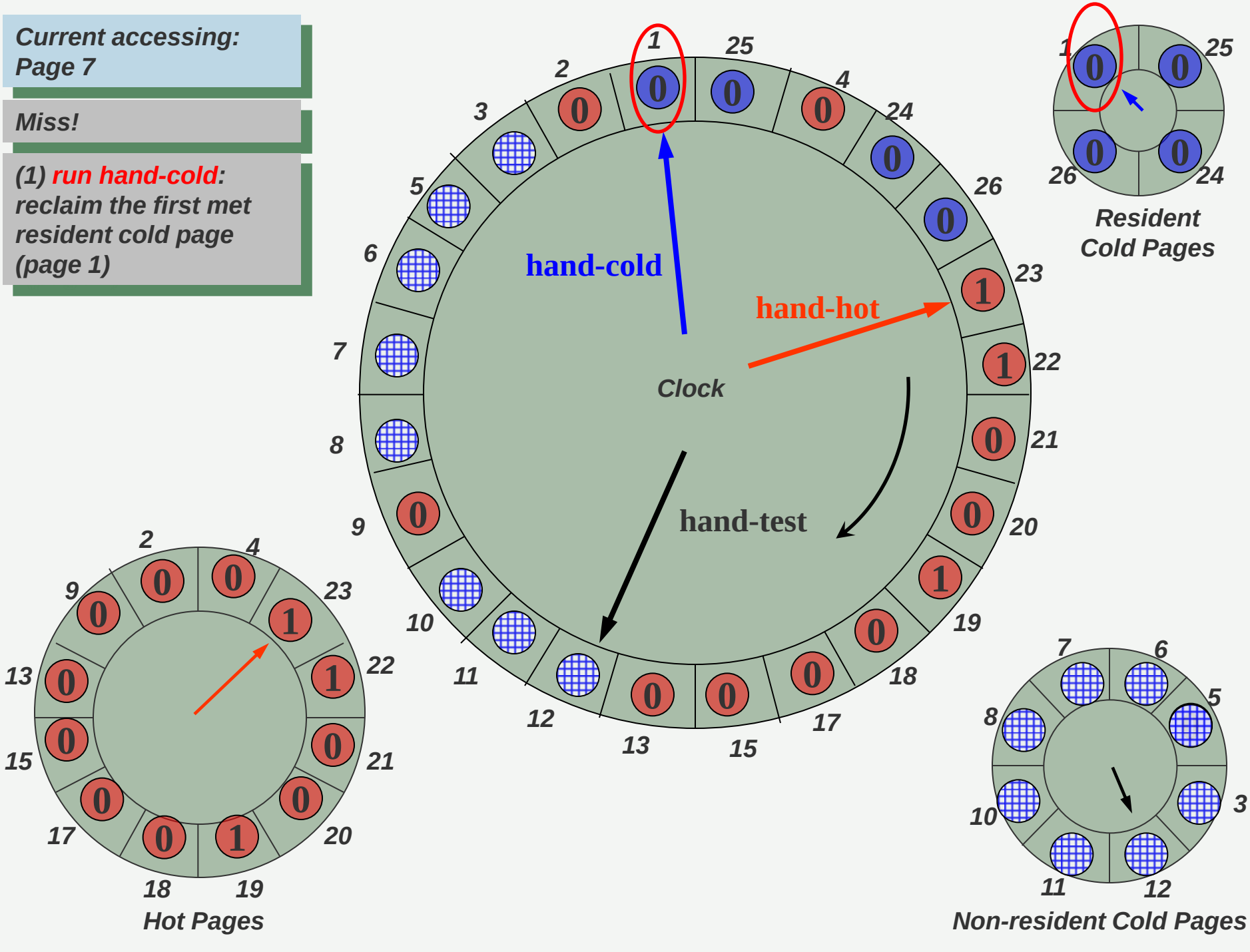
Resident Cold Pages



Hot Pages



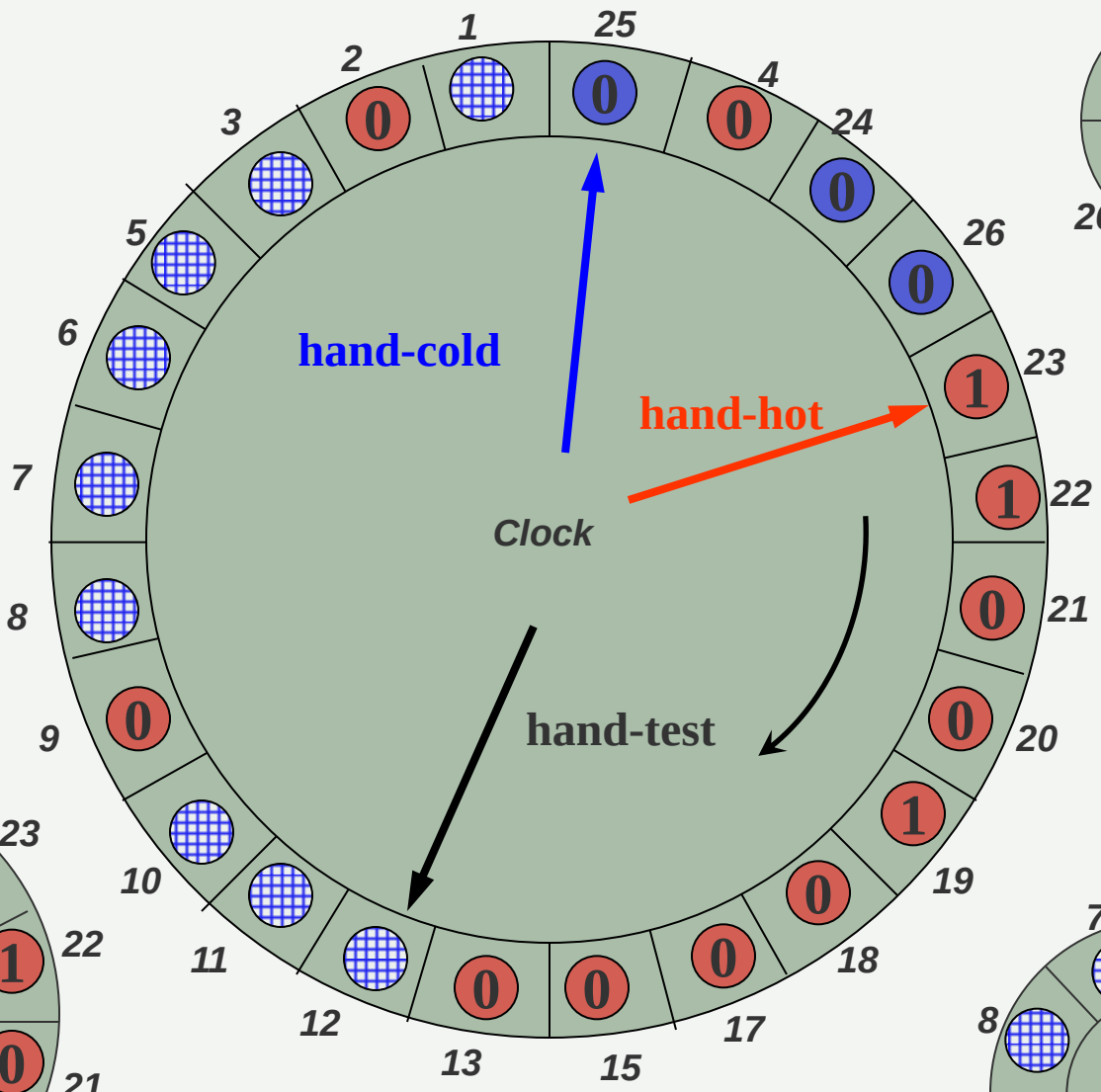
Non-resident Cold Pages

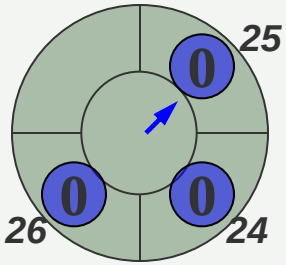

Current accessing:
Page 7
Miss!
(1) run hand-cold:
reclaim the first met
resident cold page
(page 1)
hand-cold
hand-hot
hand-test
Clock
Resident
Cold Pages
Hot Pages
Non-resident Cold Pages

Current accessing: Page 7

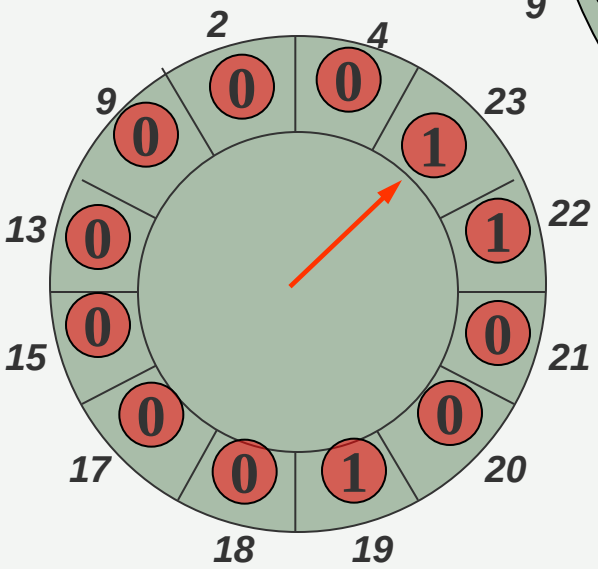
1. Leave page 1 in the old position in clock list.

2. run hand-test: to find a position for page 1

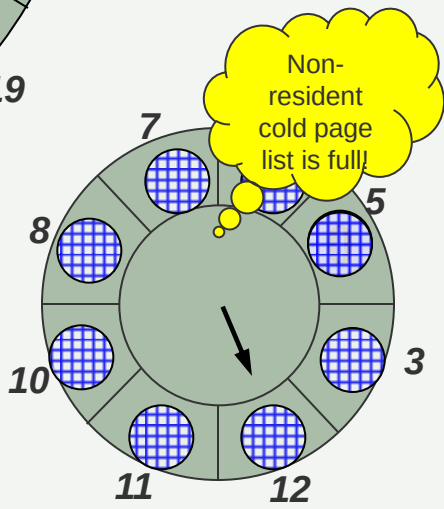




Resident Cold Pages



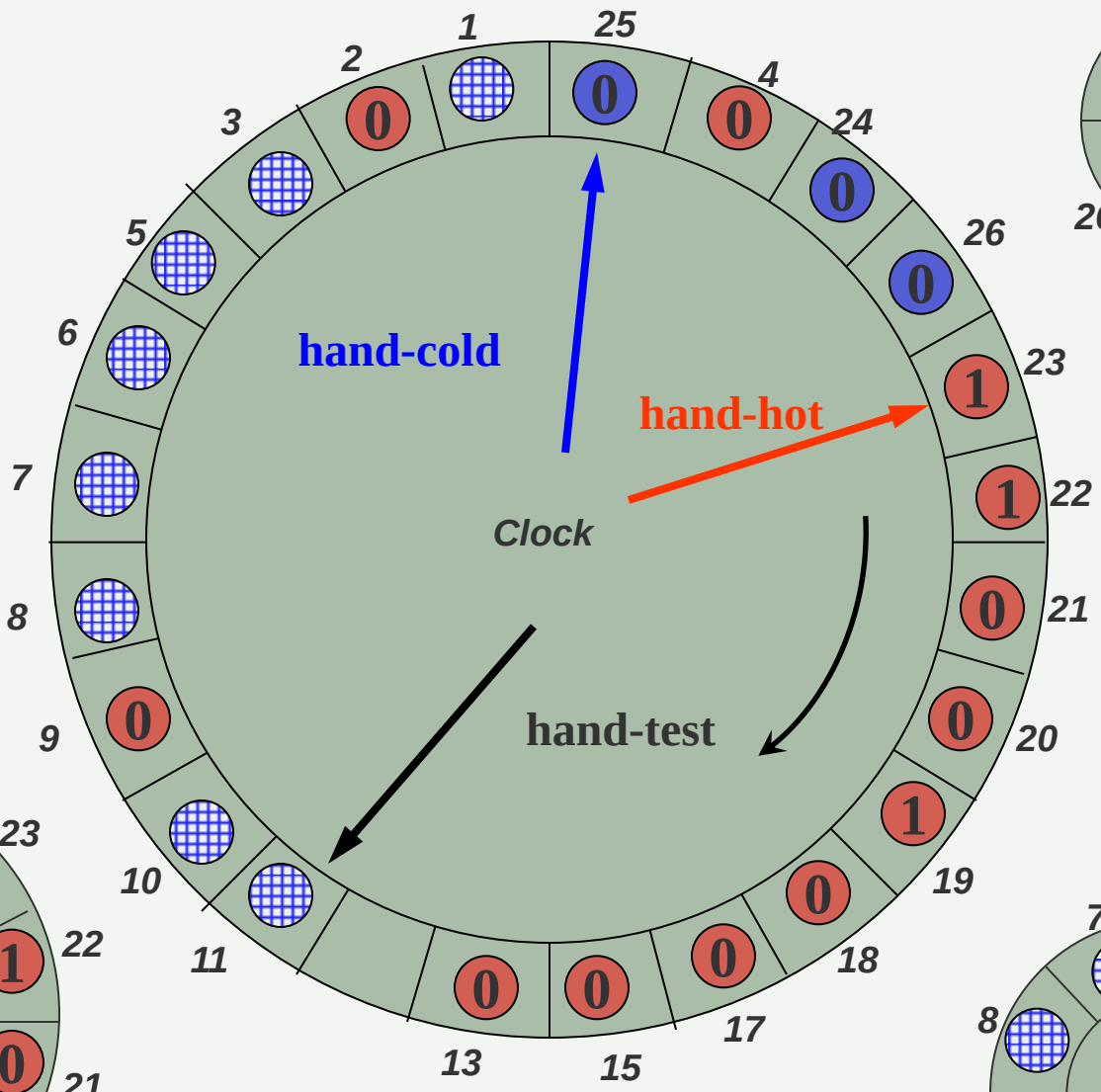
Hot Pages

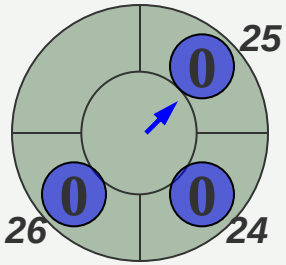


Non-resident Cold Pages

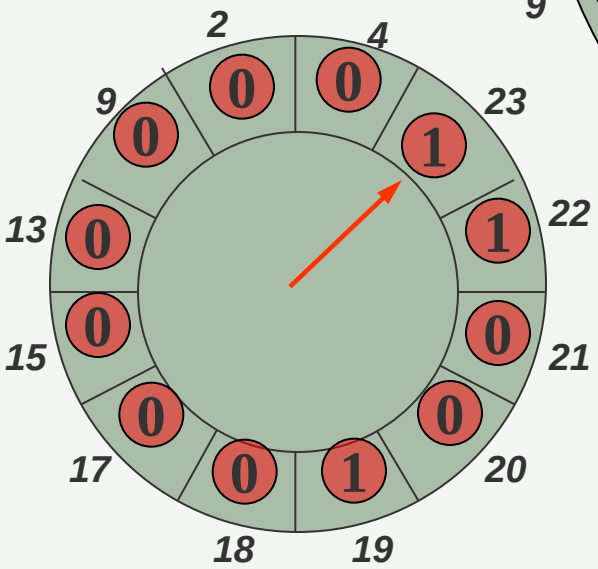
Current accessing: Page 7

3. Add page 1 into non-resident cold page list.

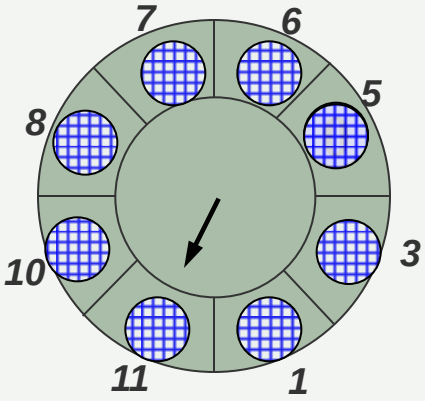




Resident Cold Pages



Hot Pages



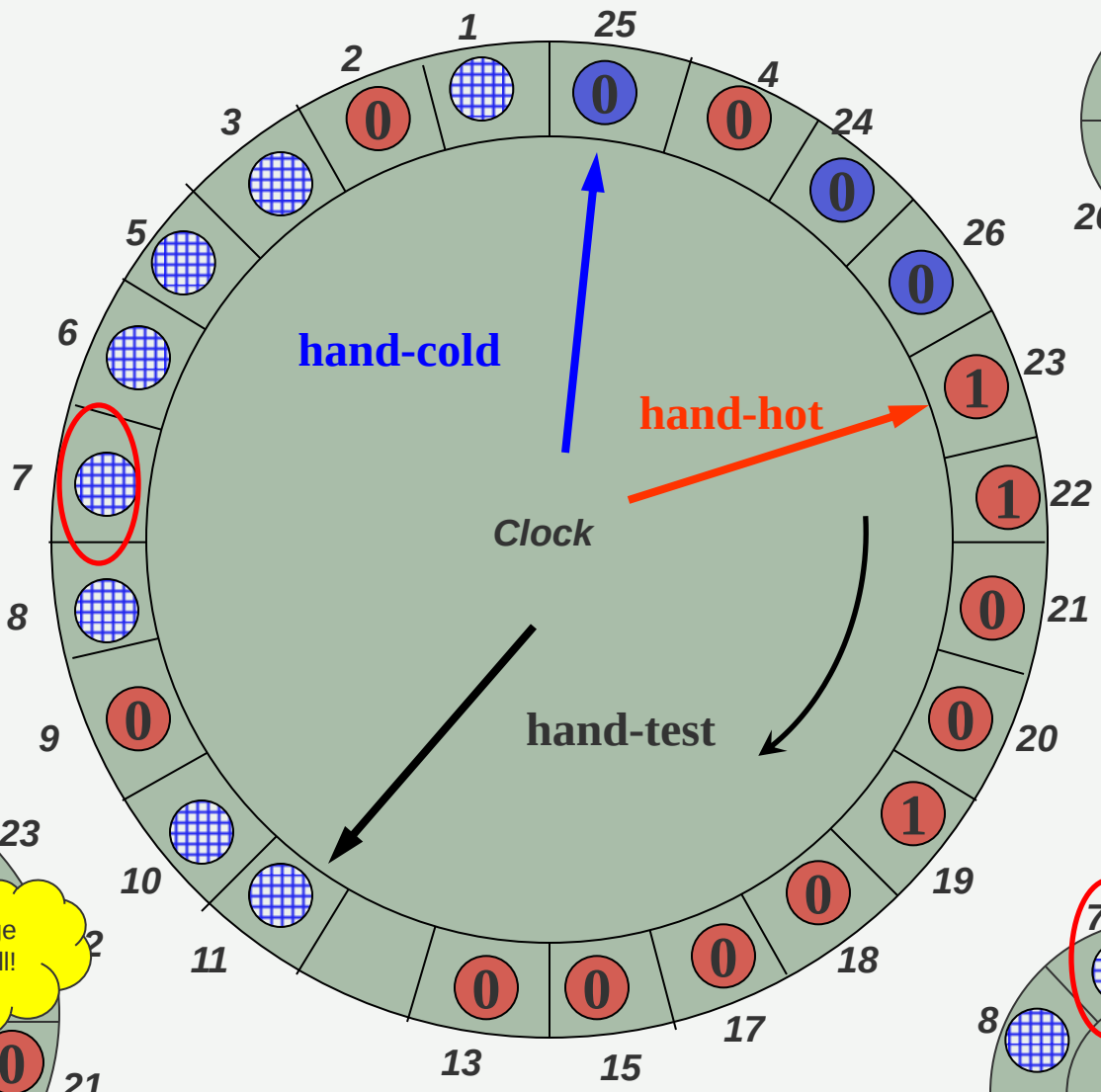
Non-resident Cold Pages

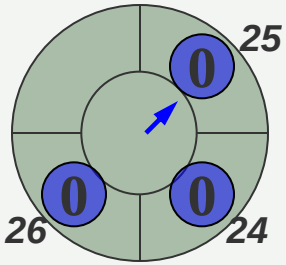
Current accessing: Page 7

Now we have one empty slot in memory

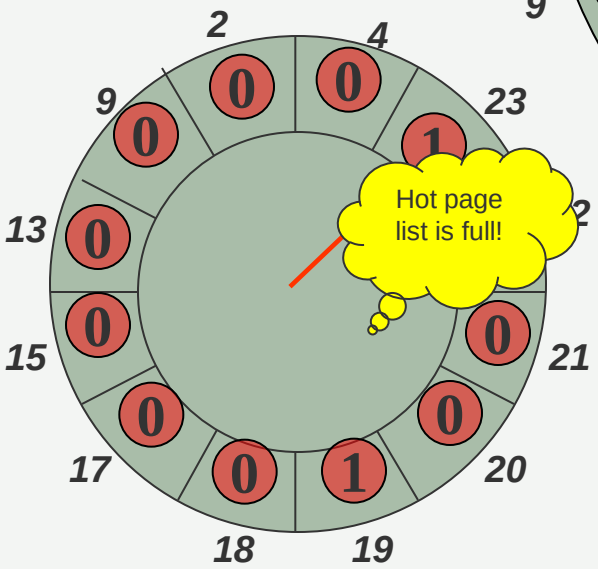
(1) check non-resident cold page list and find it---we need to upgrade it to hot page

(2) run hand-hot: degrade the first met hot page with reference bit 0 to cold page

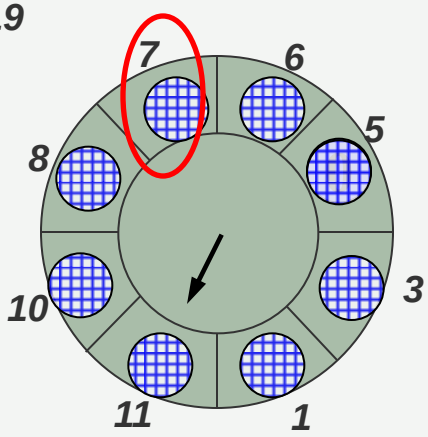




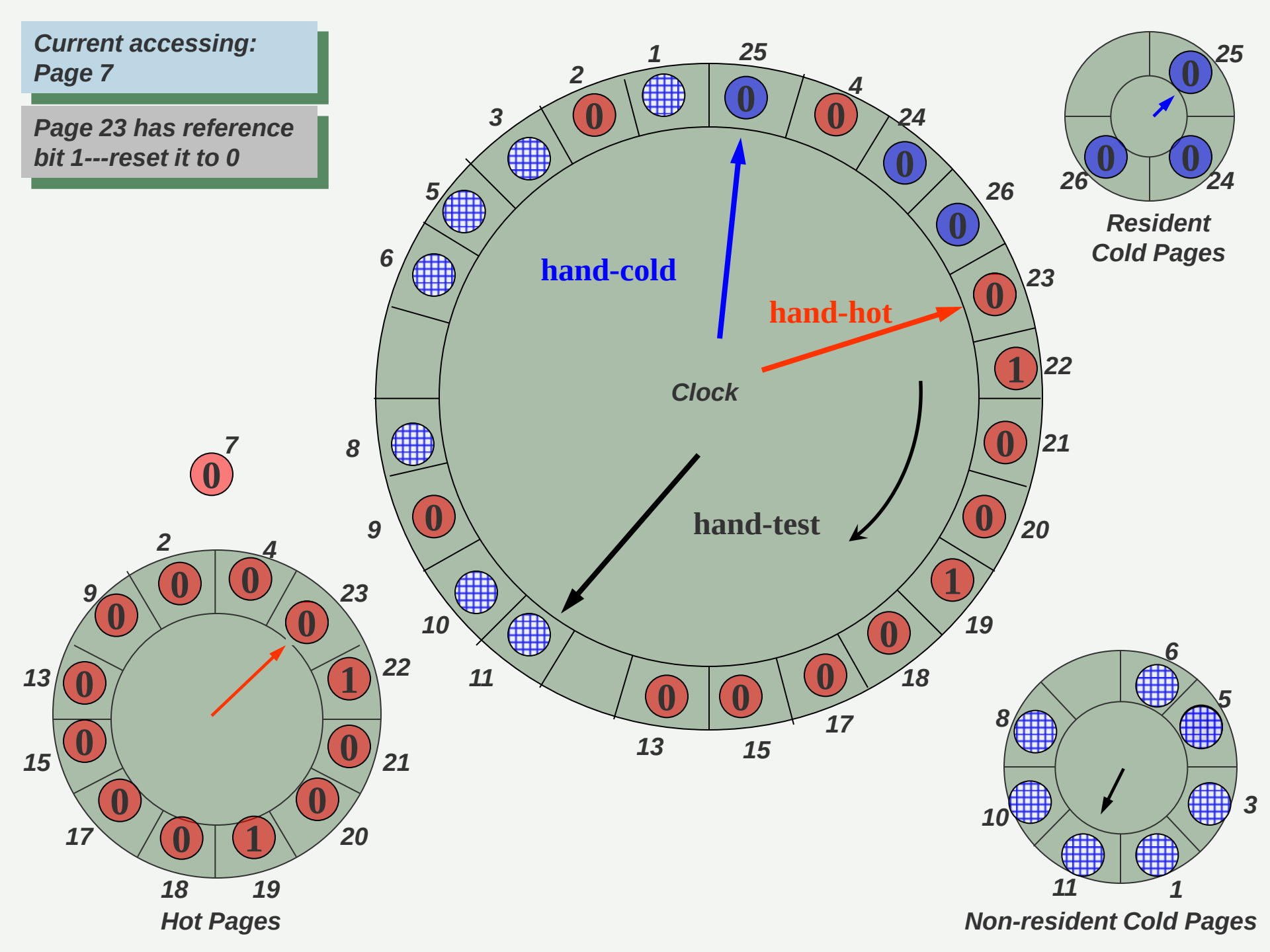
Resident Cold Pages

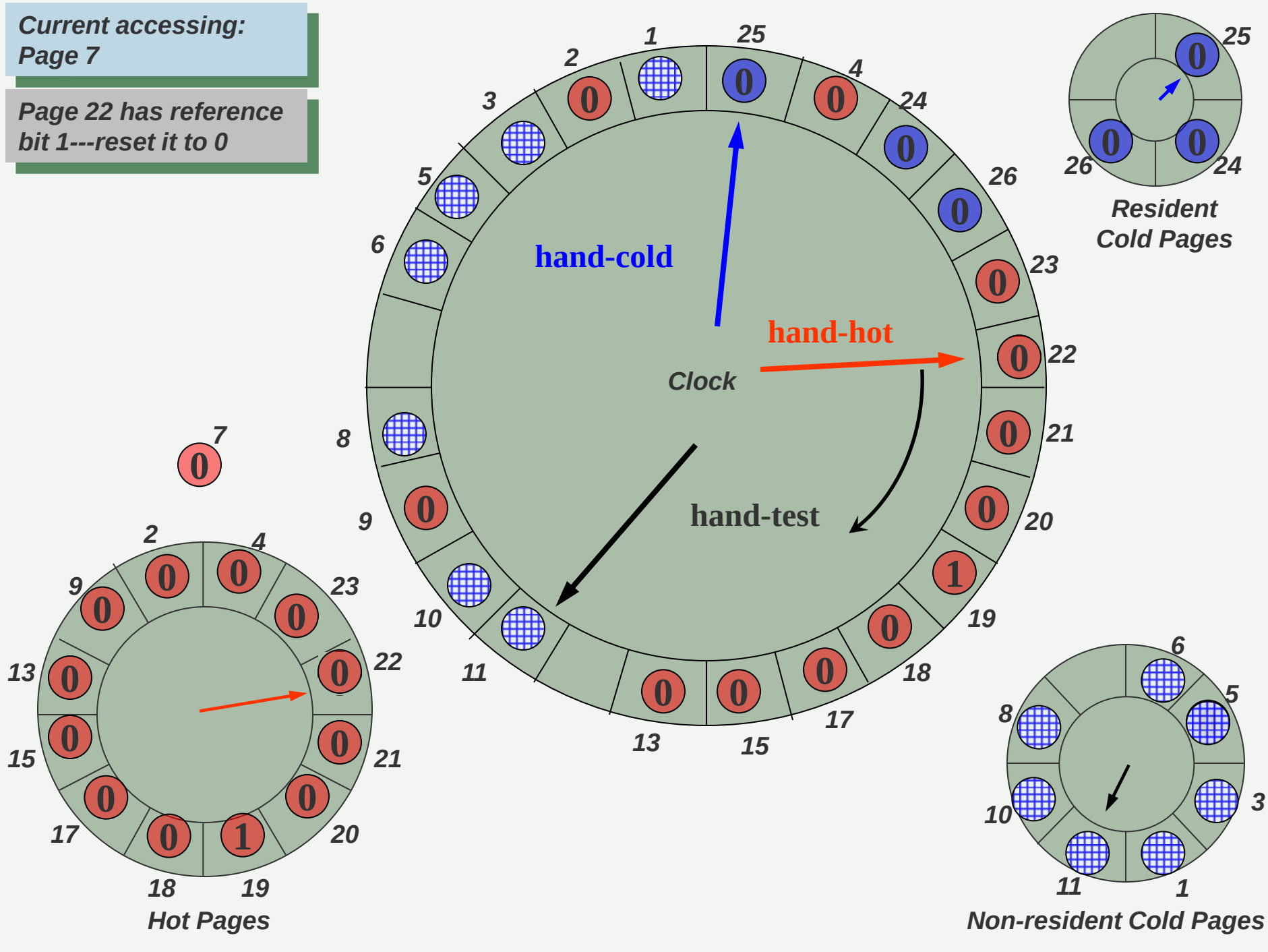


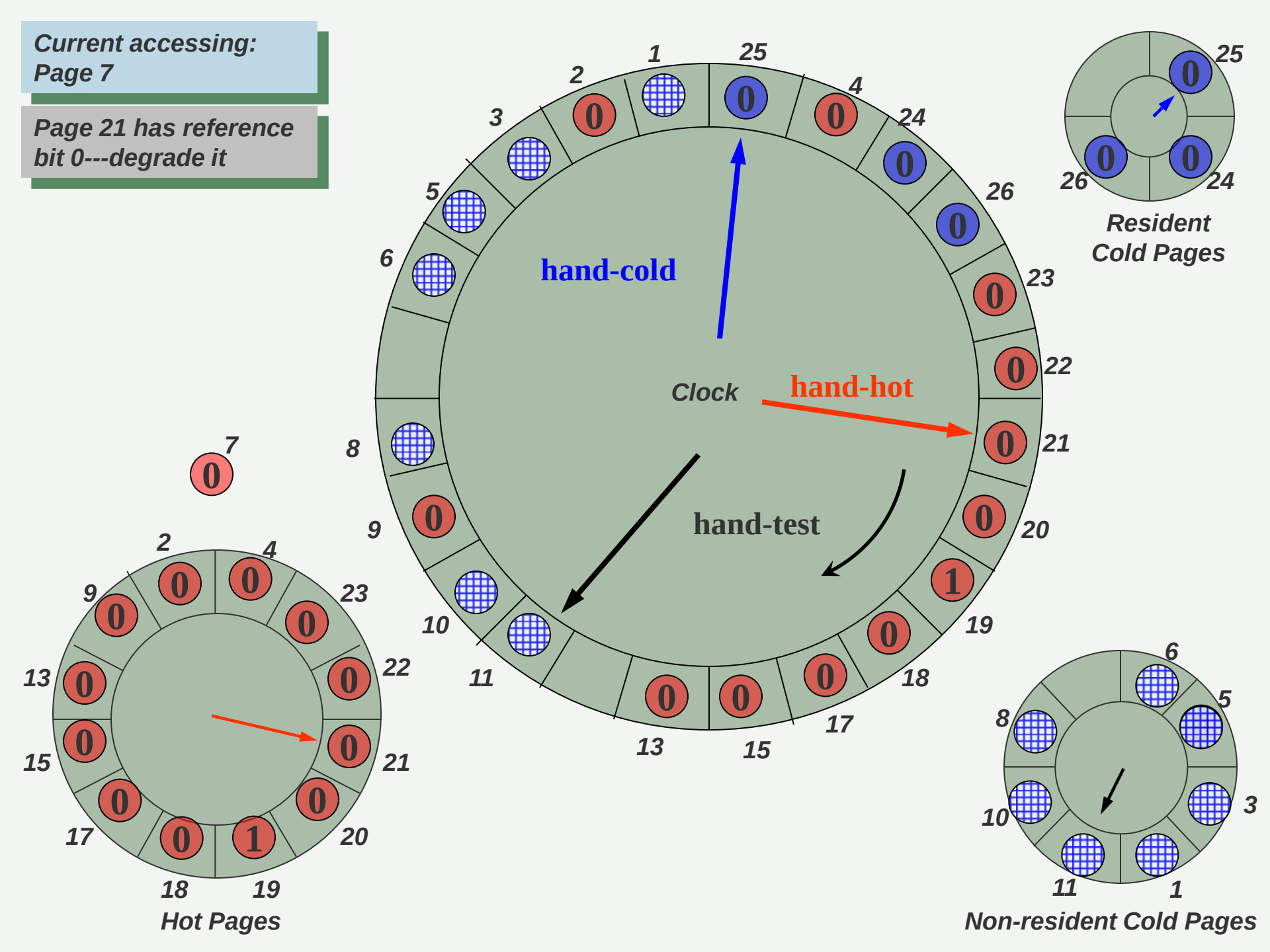
Hot Pages



Non-resident Cold Pages


Current accessing:
Page 7
Page 23 has reference bit 1---reset it to 0
hand-cold
hand-hot
hand-test
Clock
Resident Cold Pages
Hot Pages
Non-resident Cold Pages

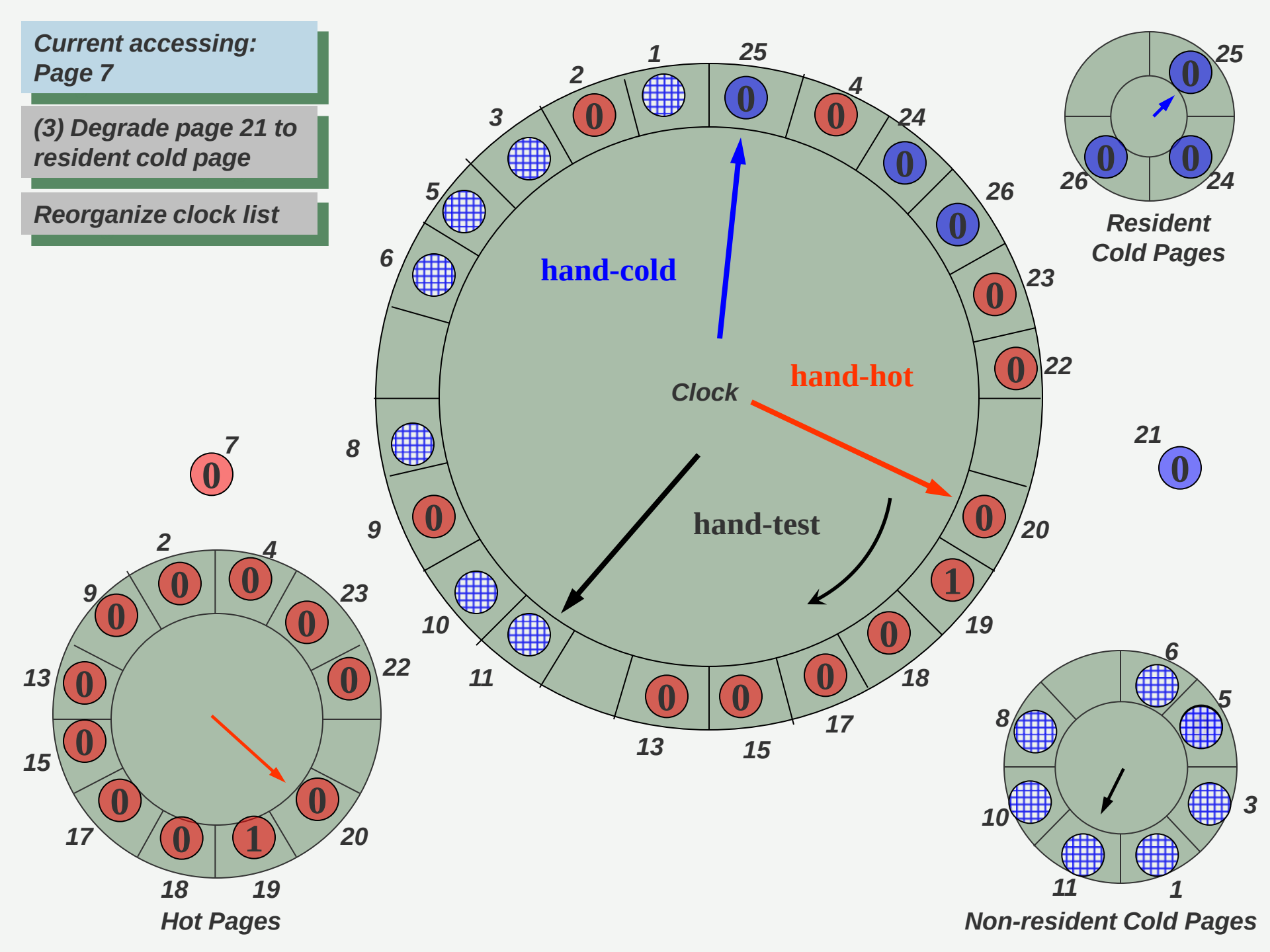

Current accessing:
Page 7
Page 22 has reference bit 1---reset it to 0
hand-cold
hand-hot
hand-test
Clock
Resident Cold Pages
Hot Pages
Non-resident Cold Pages


Current accessing:
Page 7
Page 21 has reference bit 0---degrade it
hand-cold
hand-hot
hand-test
Clock
Resident Cold Pages
Hot Pages
Non-resident Cold Pages

Current accessing: Page 7

(3) Degrade page 21 to resident cold page

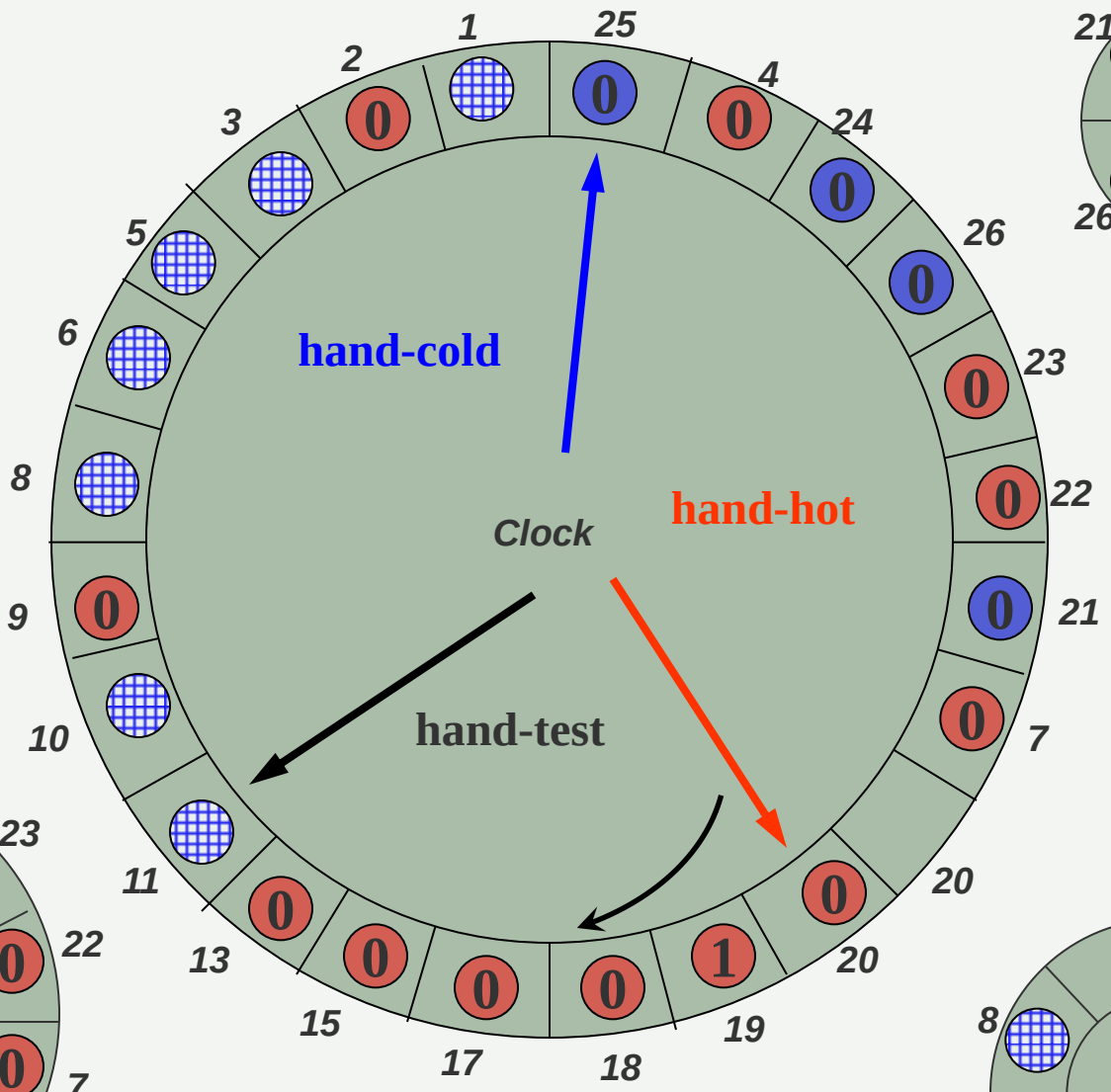
Reorganize clock list

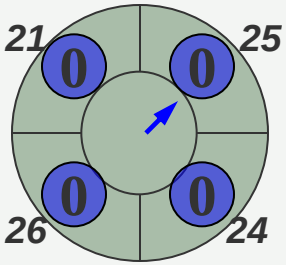


Current accessing: Page 7

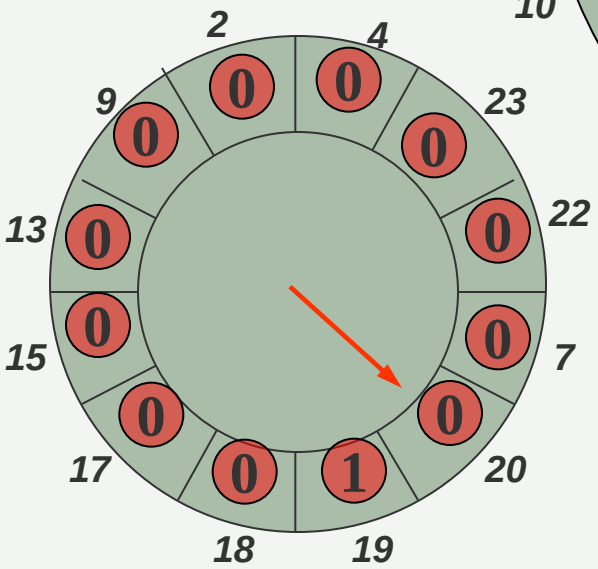
(4) insert page 21 to resident cold page and clock list

(5) insert page 7 to hot page and clock list

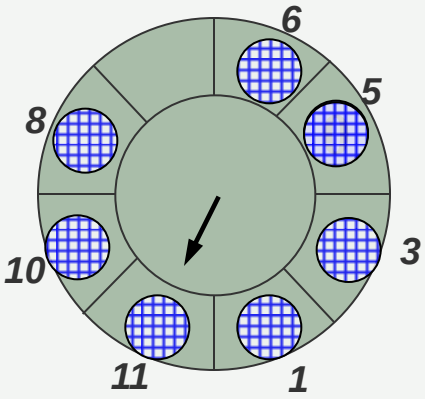




Resident Cold Pages



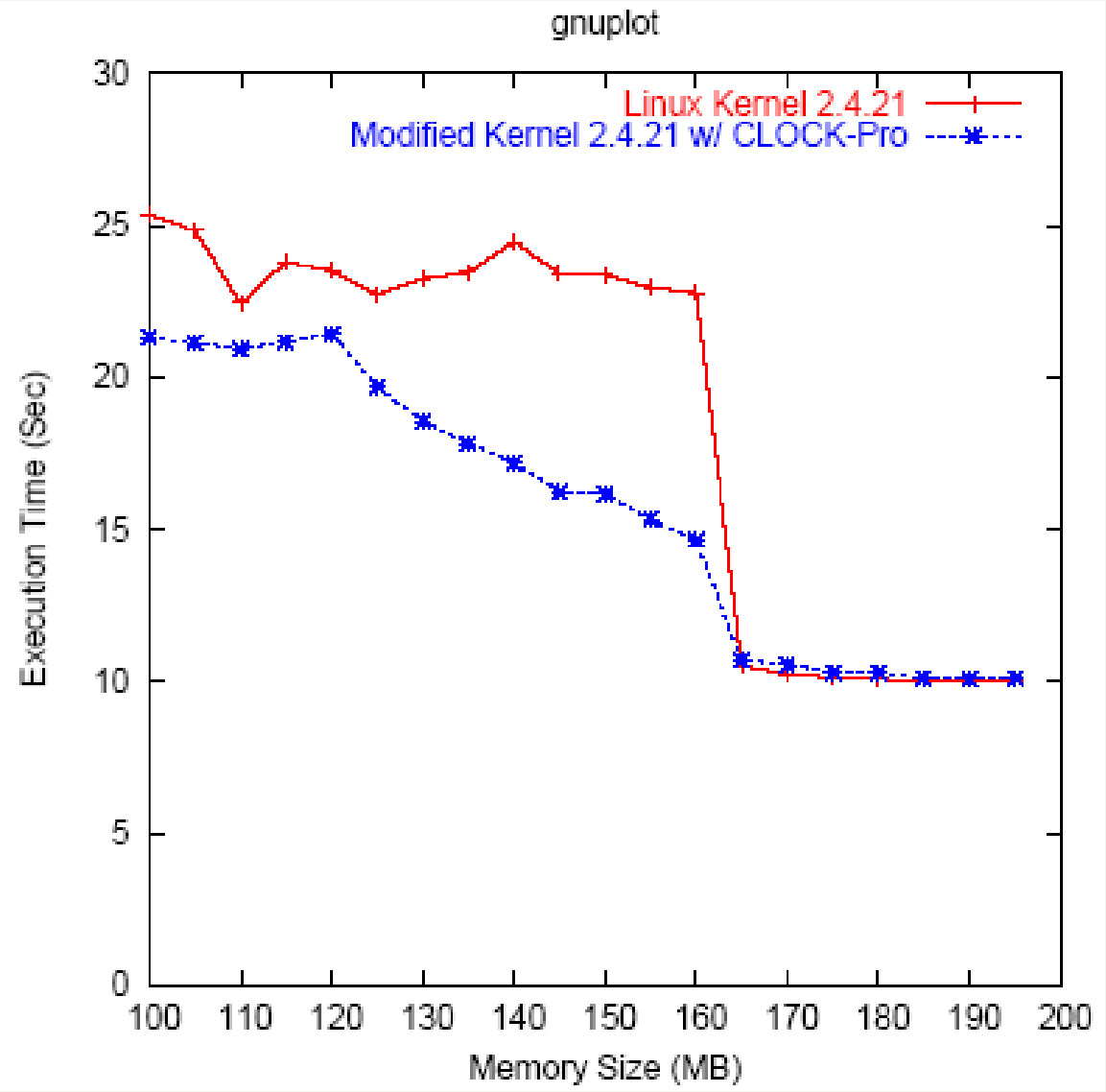
Hot Pages

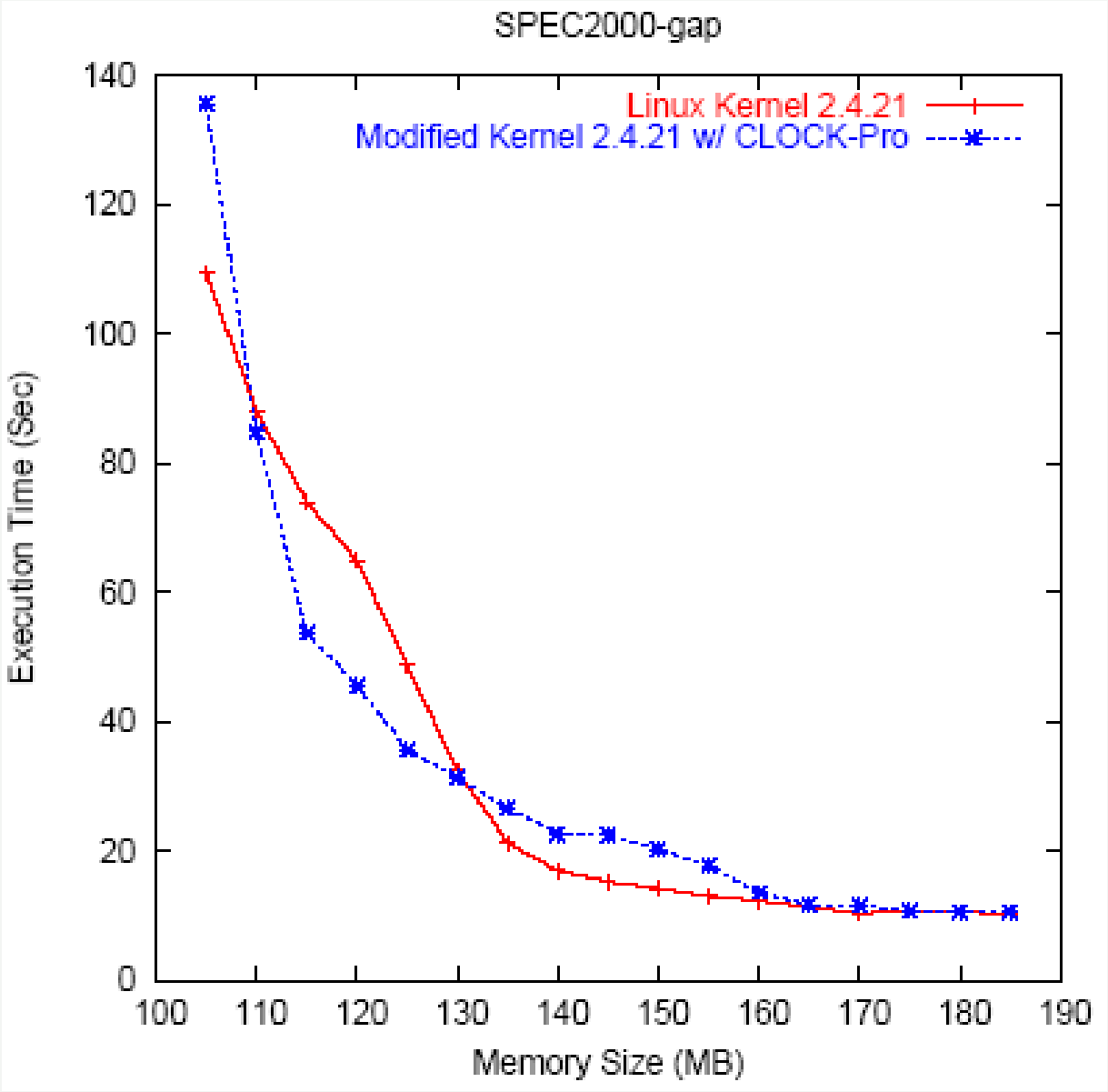


Non-resident Cold Pages

Clock-Pro Implementation in Kernels

- The Linux kernel for our implementation is 2.4.21.
  - The VM management is well documented. (a Prentice Hall book in 04, Mel Gorman)
- We are able to adjust the memory size available to the system and to the user in our experiment environment.
- All pages are placed in a single clock list in CLOCK-PRO implementation with three hands.
- SPEC 2000 and memory intensive software tools are used as benchmarks to test the CLOCK-Pro.
- Compare the modified kernel with the original.


gnuplot
Linux Kernel 2.4.21
Modified Kernel 2.4.21 w/ CLOCK-Pro
Execution Time (Sec)
Memory Size (MB)
0
5
10
15
20
25
30
100
110
120
130
140
150
160
170
180
190
200


SPEC2000-gap
Linux Kernel 2.4.21
Modified Kernel 2.4.21 w/ CLOCK-Pro
Execution Time (Sec)
Memory Size (MB)
0
20
40
60
80
100
120
140
100
110
120
130
140
150
160
170
180
190

Impact of Clock-Pro in OS and Other Systems

- Clock-pro has been adopted in NetBSD (open source Unix)
- Two Linux patches for competing its inclusion
  - Clock-pro patches in 2.6.12 by Rik van Riel
  - PeterZClockPro2 in 2.6.15-17 by Peter Zijlstra
- Clock-pro is patched in Aparche Derby (a relational DB)
- Clock-pro is patched in OpenLDAP (directory accesses)