CS 1550

Week 10

–

Project 3

Teaching Assistant

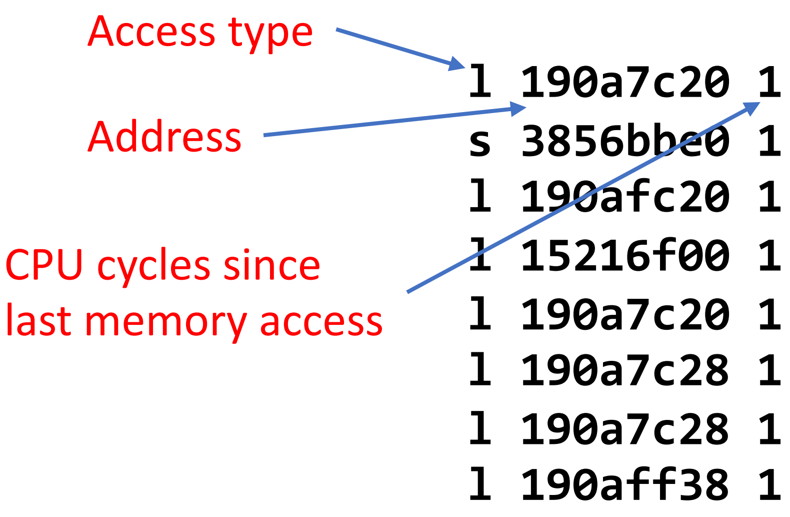
Maher Khan

Project 3 - Virtual Memory Simulator

- No need to use qemu
- You will write the simulator from scratch with Java, c++,Perl, or Python
- Read from memory traces text files
- Count the number of events (pagefaults, page evictions, hits etc.)
    - Compare eviction algorithms

Project 3 - Virtual Memory Simulator

- Simulate memory page allocation and page eviction algorithm
  - Your program will read from a memory trace
  - You will implement how loaded pages are evicted



Access type

Address

CPU cycles since last memory access

New Format:

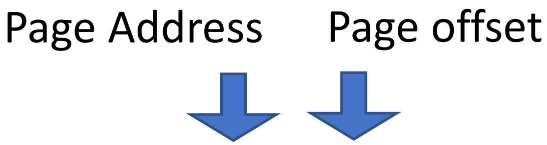
```
l 190a7c20 1
s 3856bbe0 1
l 190afc20 1
l 15216f00 1
l 190a7c20 1
l 190a7c28 1
l 190a7c28 1
l 190aff38 1
```

Old Format:

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 - Virtual Memory Simulator

- Since it is a 32-bit address space.
  - First 20 bits is used for the address
  - The rest is used for offset



```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 - Virtual Memory Simulator

- Lets suppose you have 12KB of physical memory
  - Page has 4KB
  - Assume FIFO

0	
1	
2	

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 - Virtual Memory Simulator

- Lets suppose you have 12KB of physical memory
  - Page has 4KB
  - Assume FIFO

0	
1	
2	

Pagefault since it is not in the process table

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 - Virtual Memory Simulator

- Lets suppose you have 12KB of physical memory
  - Page has 4KB
  - Assume FIFO

0	190a7
1	
2	

Pagefault since it is not in the process table

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 - Virtual Memory Simulator

- Lets suppose you have 12KB of physical memory
  - Page has 4KB
  - Assume **FIFO**

We need to evict someone!!

0	**190a7**
1	**3856b**
2	**190af**

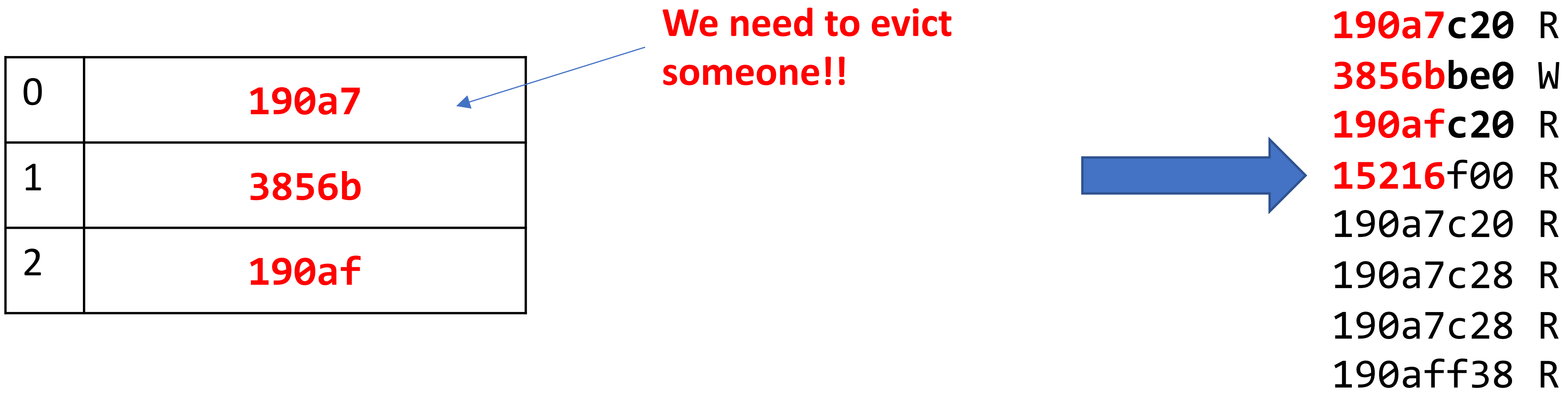
Pagefault again

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 - Virtual Memory Simulator

- Lets suppose you have 12KB of physical memory
  - Page has 4KB
  - Assume **FIFO**

Pagefault again



0	**190a7**
1	**3856b**
2	**190af**

We need to evict someone!!

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 - Virtual Memory Simulator

- Lets suppose you have 12KB of physical memory
  - Page has 4KB
  - Assume **FIFO**

Pagefault again

0	3856b
1	190af
2	

We need to evict someone!!

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 - Virtual Memory Simulator

- You have to implement:
  - Opt
  - FIFO
  - Aging

Project 3 – Optimal algorithm

- Evicts the page that will not be used the longest in the future.

Project 3 – Optimal algorithm

- Evicts the page that will not be used the longest in the future.

Pagefault again

0	190a7
1	3856b
2	190af

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page that will not be used the longest in the future.

Pagefault again

We need to evict someone!!

0	**190a7**
1	**3856b**
2	**190af**

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future.**

Pagefault again

We need to evict someone!!

0	**190a7**
1	**3856b**
2	**190af**

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

Let’s analyze who will be needed furthest away in the trace

0	**190a7**	
1	**3856b**	
2	**190af**	

We need to evict someone!!

Pagefault again

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

Pagefault again

We need to evict someone!!

0	190a7	
1	3856b	
2	190af	

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

Pagefault again

We need to evict someone!!

0	**190a7**	0
1	**3856b**	
2	**190af**	

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

Pagefault again

We need to evict someone!!

0	**190a7**	0
1	**3856b**	4
2	**190af**	

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

Pagefault again

We need to evict someone!!

0	**190a7**	0
1	**3856b**	4
2	**190af**	3

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

0	190a7	0
1	3856b	4
2	190af	3

We need to evict someone!!

Pagefault again

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

Pagefault again

0	**190a7**	0
1		
2	**190af**	3

We need to evict someone!!

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

Pagefault again

0	**190a7**	0
1		
2	**190af**	3

We need to evict someone!!

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

Pagefault again

0	**190a7**	0
1	**15216**	
2	**190af**	3

We need to evict someone!!

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – Optimal algorithm

- Evicts the page **that will not be used the longest in the future**.

Pagefault again

0	**190a7**	0
1	**15216**	
2	**190af**	3

Remember that this will change as the memory trace progresses

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
3856bbe0 R
```

Project 3 – First In First Out(FIFO)

- Evicts the oldest page in memory.

Project 3 – First In First Out(FIFO)

- Evicts the oldest page in memory.

0	190a7
1	3856b
2	190af

Pagefault again

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 – First In First Out(FIFO)

- Evicts the oldest page in memory.

Pagefault again

0	**190a7**
1	**3856b**
2	**190af**

We need to evict someone!!

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 – First In First Out(FIFO)

- Evicts the oldest page in memory.

Pagefault again

0	3856b
1	190af
2	

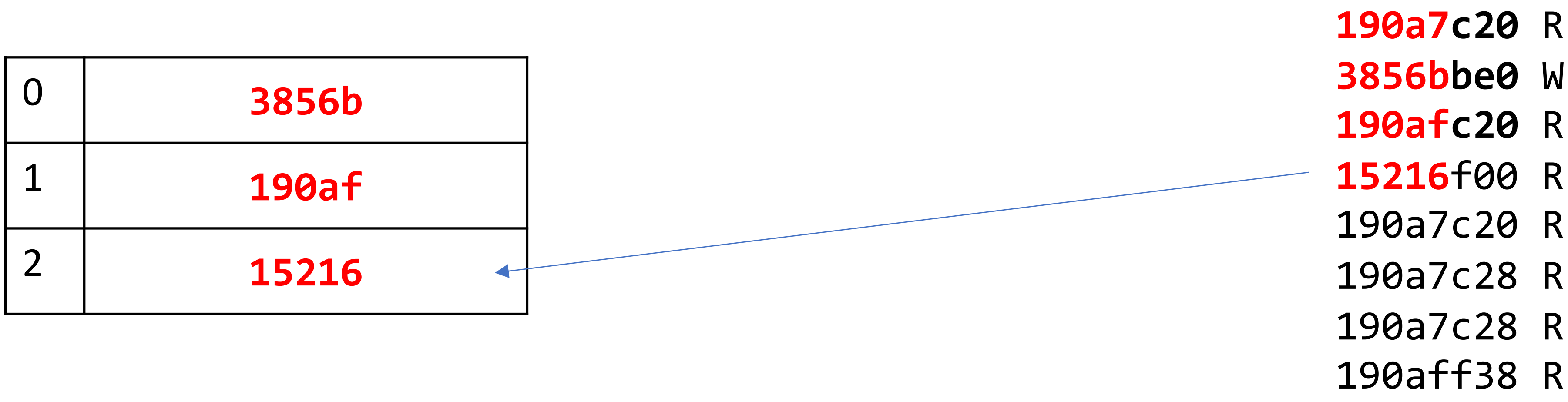
We need to evict someone!!

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 – First In First Out(FIFO)

- Evicts the oldest page in memory.

Pagefault again



0	**3856b**
1	**190af**
2	**15216**

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 – First In First Out(FIFO)

- Evicts the oldest page in memory.

Pagefault again

0	**3856b**
1	**190af**
2	**15216**

Next to be evicted

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 – Aging

- Evicts pages that are not being used.
  - Periodically changes the “clock” counter.
  - Every page fault, update a timestamp of when the page was inserted
  - Pages are ranked according to counter

0	**190a7**
1	**3856b**
2	**190af**

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 – Aging

- Evicts pages that are not being used.
  - Select page to be evicted by finding the lowest counter value
  - On reference, set leftmost bit of a counter (can be done by copying the reference bit to the counter at the clock tick)

	Referenced	Counter	
0	1	1000000	**190a7**
1	1	1000000	**3856b**
2	1	?	**190af**

```
190a7c20 R  0
3856bbe0 W  6
190afc20 R 10
15216f00 R 11
190a7c20 R 14
190a7c28 R 15
190a7c28 R 16
190aff38 R 22
```

Project 3 – Aging

- Assume refresh interval is 10
  - Select page to be evicted by finding the lowest counter value
  - On reference, set leftmost bit of a counter (can be done by copying the reference bit to the counter at the clock tick)

	Referenced	Counter	
0	1	0100000	**190a7**
1	1	0100000	**3856b**
2	1	1000000	**190af**

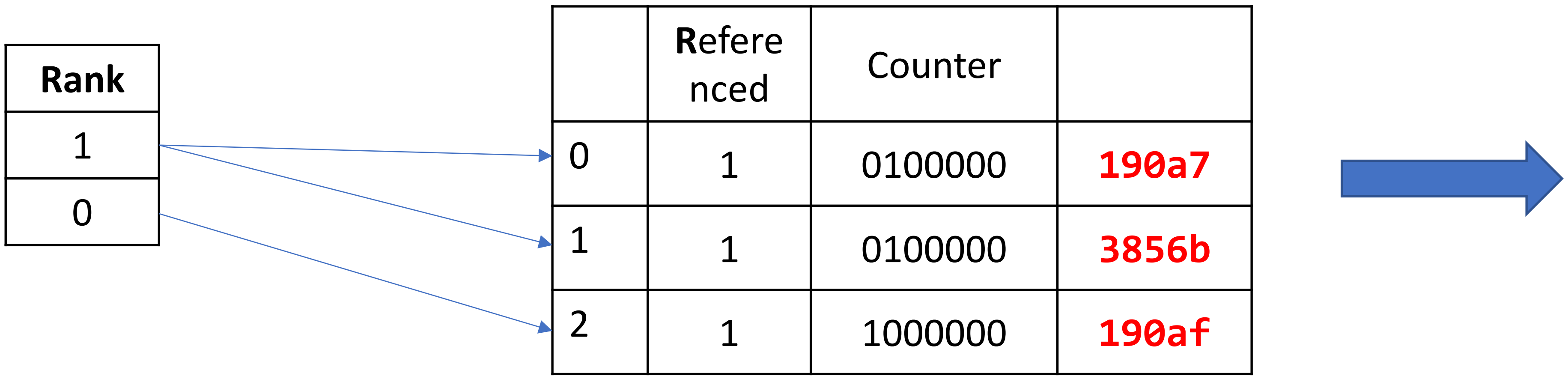
```
190a7c20 R  0
3856bbe0 W  6
190afc20 R 10
15216f00 R 11
190a7c20 R 14
190a7c28 R 15
190a7c28 R 16
190aff38 R 22
```

Project 3 – Aging

- New page fault
  - Rank pages to be evicted by finding the lowest counter value

We need to evict someone.

But there are 2 pages at Rank 1 !!



Rank
1
0

	Referenced	Counter	
0	1	0100000	**190a7**
1	1	0100000	**3856b**
2	1	1000000	**190af**

```
190a7c20 R  0
3856bbe0 W  6
190afc20 R 10
15216f00 R 11
190a7c20 R 14
190a7c28 R 15
190a7c28 R 16
190aff38 R 22
```

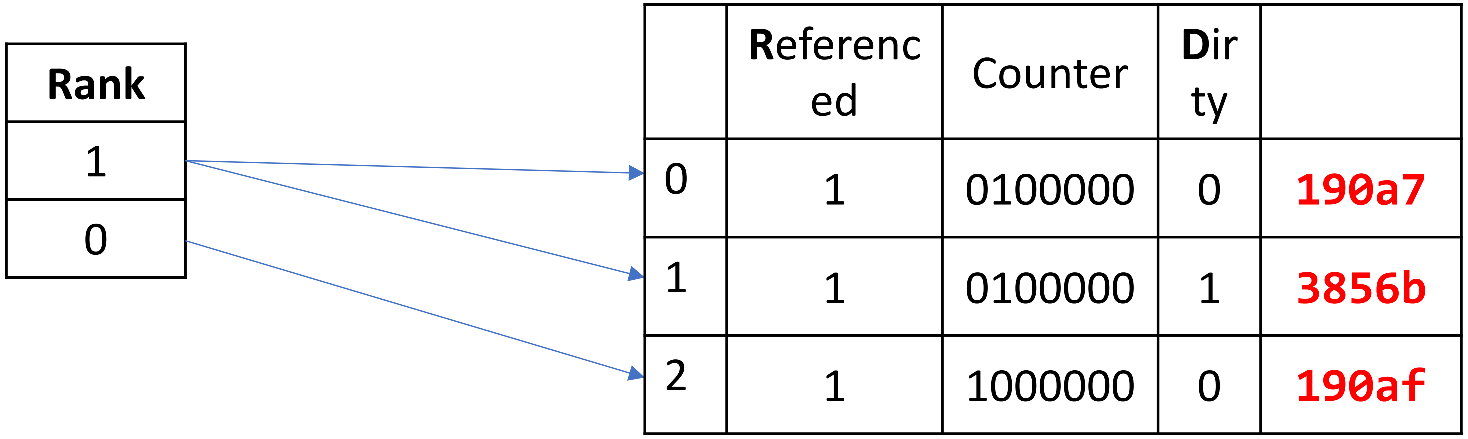
Project 3 – Aging

- New page fault
  - Rank pages to be evicted by finding the lowest counter value

We need to evict someone.

But there are 2 pages at Rank 1 !!

Which is the best to evict??



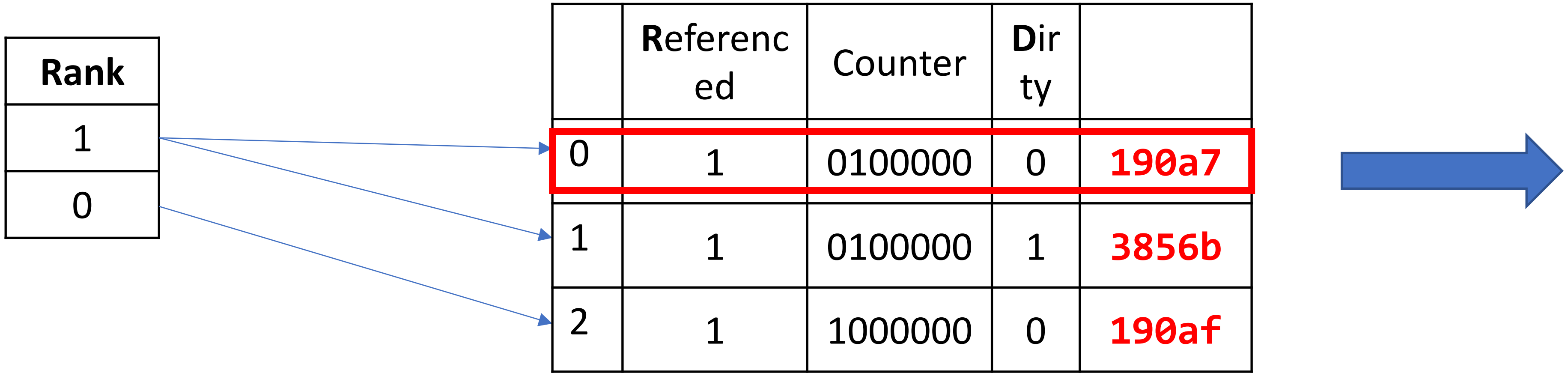
Rank
1
0

	Referenced	Counter	**Dirty**	
0	1	0100000	0	**190a7**
1	1	0100000	1	**3856b**
2	1	1000000	0	**190af**

```
190a7c20 R  0
3856bbe0 W  6
190afc20 R 10
15216f00 R 11
190a7c20 R 14
190a7c28 R 15
190a7c28 R 16
190aff38 R 22
```

Project 3 – Aging

- New page fault
  - Rank pages to be evicted by finding the lowest counter value
  - Can consider counter value as counter || dirty



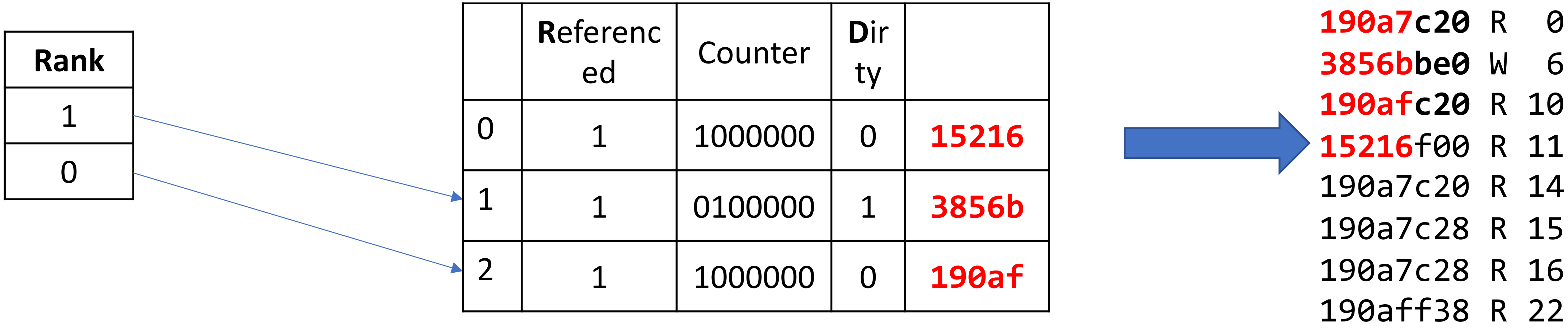
Rank
1
0

	Referenced	Counter	Dirty	
0	1	0100000	0	190a7
1	1	0100000	1	3856b
2	1	1000000	0	190af

```
190a7c20 R  0
3856bbe0 W  6
190afc20 R 10
15216f00 R 11
190a7c20 R 14
190a7c28 R 15
190a7c28 R 16
190aff38 R 22
```

Project 3 – Aging

- New page fault
  - Evict and add the new page



Rank
1
0

	Referenced	Counter	Dirty	
0	1	1000000	0	15216
1	1	0100000	1	3856b
2	1	1000000	0	190af

```
190a7c20 R  0
3856bbe0 W  6
190afc20 R 10
15216f00 R 11
190a7c20 R 14
190a7c28 R 15
190a7c28 R 16
190aff38 R 22
```

Project 3 – Program interface

- Program UI

```
./vmsim –n <numframes> -a <opt|aging|fifo> [-r <refresh>] <tracefile>
```

Project 3 – Program interface

- Program UI

```
./vmsim –n <numframes> -a <opt|aging|fifo> [-r <refresh>] <tracefile>
```

Specifies the number of Memory slots.

Project 3 – Program interface

- Program UI

```
./vmsim –n <numframes> -a <opt|aging|fifo> [-r <refresh>] <tracefile>
```

Specifies which algorithm to run

Project 3 – Program interface

- Program UI

```
./vmsim –n <numframes> -a <opt|aging|fifo> [-r <refresh>] <tracefile>
```

Specifies the periodicity of the refresh rate for the **aging algorithm**

Project 3 – Program interface

- Program UI

```
./vmsim –n <numframes> -a <opt|aging|fifo> [-r <refresh>] <tracefile>
```

Path to memory trace file

Project 3 – Program interface

- Program UI

```
./vmsim –n <numframes> -a <opt|aging|fifo> [-r <refresh>] <tracefile>
```

```
python vmsim.py    –n 8 -a opt -r ./swim.trace

java   vmsim.class –n 8 -a opt -r ./swim.trace
```

Project 3 – Program interface

- Program UI

```
./vmsim –n <numframes> -a <opt|aging|fifo> [-r <refresh>] <tracefile>
```

```
python vmsim.py    –n 8 -a opt -r ./swim.trace

java   vmsim.class –n 8 -a opt -r ./swim.trace
```

Project 3 – Program interface

- As the simulation runs you should print in the following format for **each memory reference**.
    - hit
    - page fault – no eviction
    - page fault – evict clean
    - page fault – evict dirty

Project 3 – Program interface

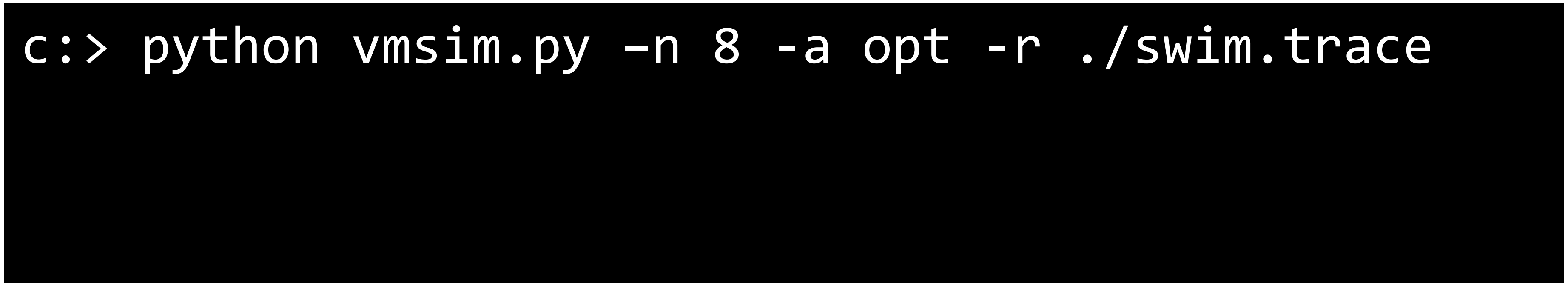
- As the simulation runs you should print in the following format for **each memory reference**.

```
C:>
```

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 – Program interface

- As the simulation runs you should print in the following format for **each memory reference**.



```
c:> python vmsim.py –n 8 -a opt -r ./swim.trace
```

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

Project 3 – Program interface

- As the simulation runs you should print in the following format for **each memory reference**.

```
c:> python vmsim.py –n 8 -a opt -r ./swim.trace

hit
page fault – no eviction
hit
page fault – evict dirty
page fault – evict clean
…
```

```
190a7c20 R
3856bbe0 W
190afc20 R
15216f00 R
190a7c20 R
190a7c28 R
190a7c28 R
190aff38 R
```

CS 1550 – Project 3

- **Due**: Friday, March 22, 2019 @11:59pm
- **Late**: Sunday, March 24, 2019 @11:59pm
  - 10% reduction per late day

CS 1550

Week 10

–

Project 3

Teaching Assistant

Maher Khan