CS 1550

Week 9 – Project 3

Teaching Assistant

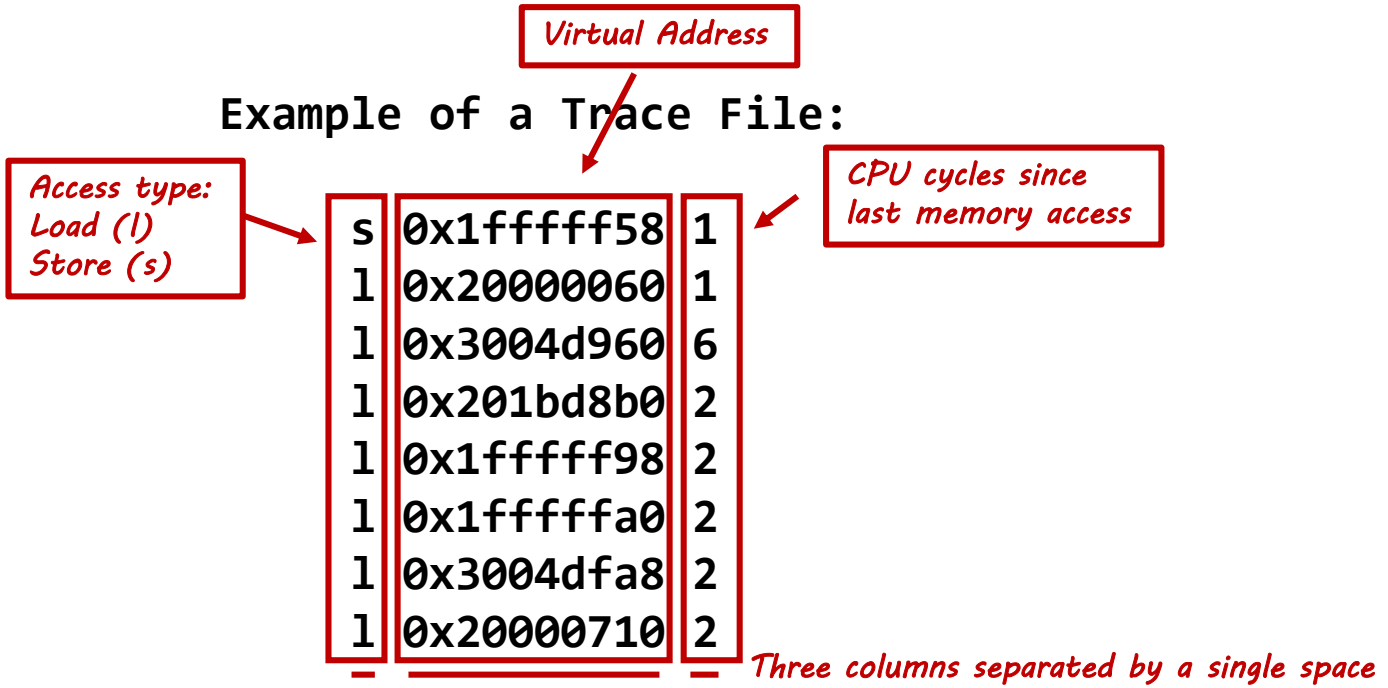
Xiaoyu(Veronica) Liang

Project 3 - Virtual Memory Simulator

- **NO** need to use QEMU
- You will write the simulator from scratch with **C/C++**, **Java**, **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



Project 3 - Virtual Memory Simulator

- 32-bit address
- 4KB page size (4K: 0x000 ~ 0xFFF)
  - First 20 bits is used for the address
  - The rest is used for offset

```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1fffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 - Virtual Memory Simulator

- 32-bit address
- 4KB page size (4K: 0x000 ~ 0xFFF)
  - First 20 bits is used for the address
  - The rest is used for offset

Project 3 - Virtual Memory Simulator

- Given 3 page frames in 4KB page size
  - Assume FIFO

0	
1	
2	

Pagefault since it is not in the process table

```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1fffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 - Virtual Memory Simulator

- Given 3 page frames in 4KB page size
  - Assume FIFO

0	1ffff
1	
2	

Pagefault since it is not in the process table

```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1ffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 - Virtual Memory Simulator

- Given 3 page frames in 4KB page size
  - Assume FIFO

0	1ffff
1	20000
2	

Pagefault

```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1ffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 - Virtual Memory Simulator

- Given 3 page frames in 4KB page size
  - Assume FIFO

0	1ffff
1	20000
2	3004d

Pagefault

```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1ffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 - Virtual Memory Simulator

- Given 3 page frames in 4KB page size
  - Assume FIFO

0	1ffff
1	20000
2	3004d

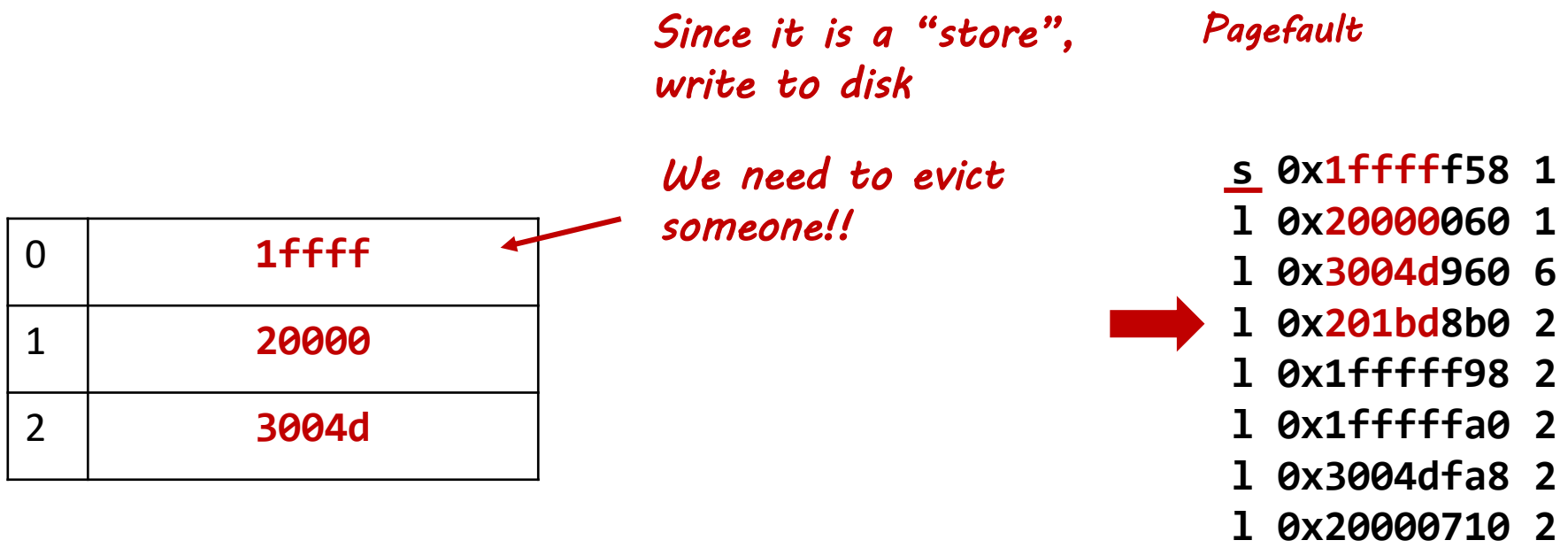
We need to evict someone!!

Pagefault

```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1ffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 - Virtual Memory Simulator

- Given 3 page frames in 4KB page size
  - Assume **FIFO**



Project 3 - Virtual Memory Simulator

- Given 3 page frames in 4KB page size
  - Assume **FIFO**

0	**20000**
1	**3004d**
2	

Pagefault

```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1ffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 - Virtual Memory Simulator

- Given 3 page frames in 4KB page size
  - Assume **FIFO**

0	**20000**
1	**3004d**
2	**201bd**

Load new page

Pagefault

```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1ffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 - Virtual Memory Simulator

- Given 3 page frames in 4KB page size
  - Assume **FIFO**

0	**20000**
1	**3004d**
2	**201bd**

Next to be evicted, if need

Pagefault

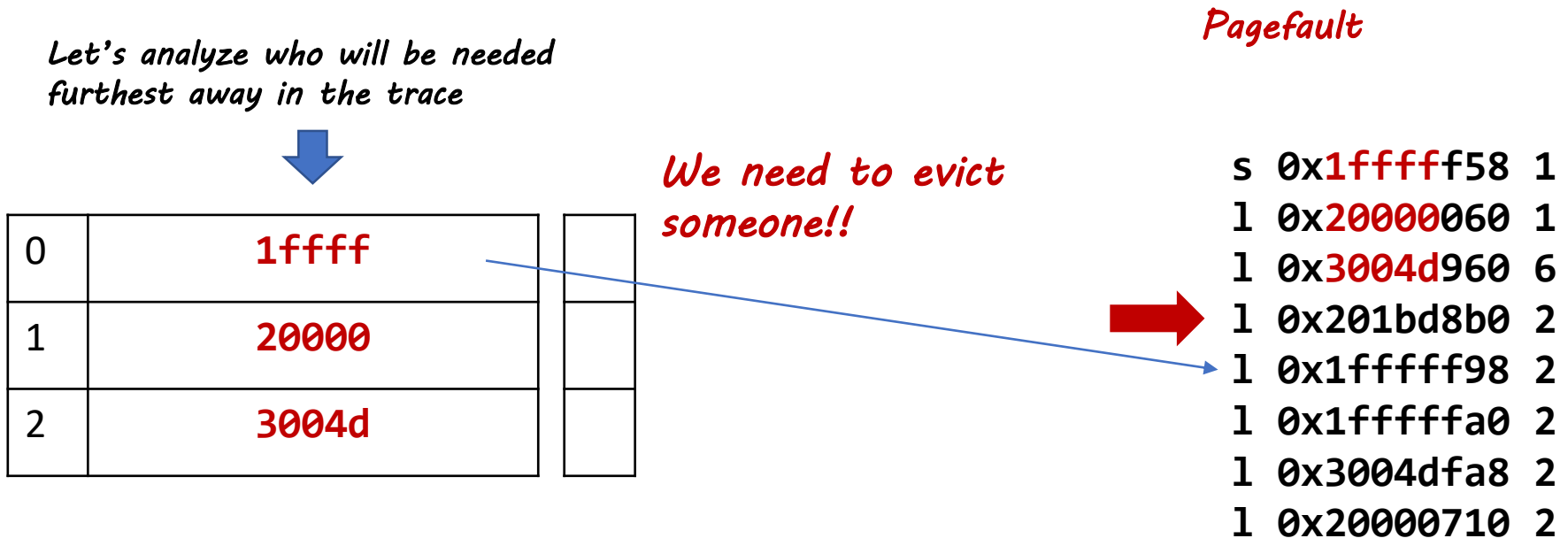
```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1ffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 - Virtual Memory Simulator

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

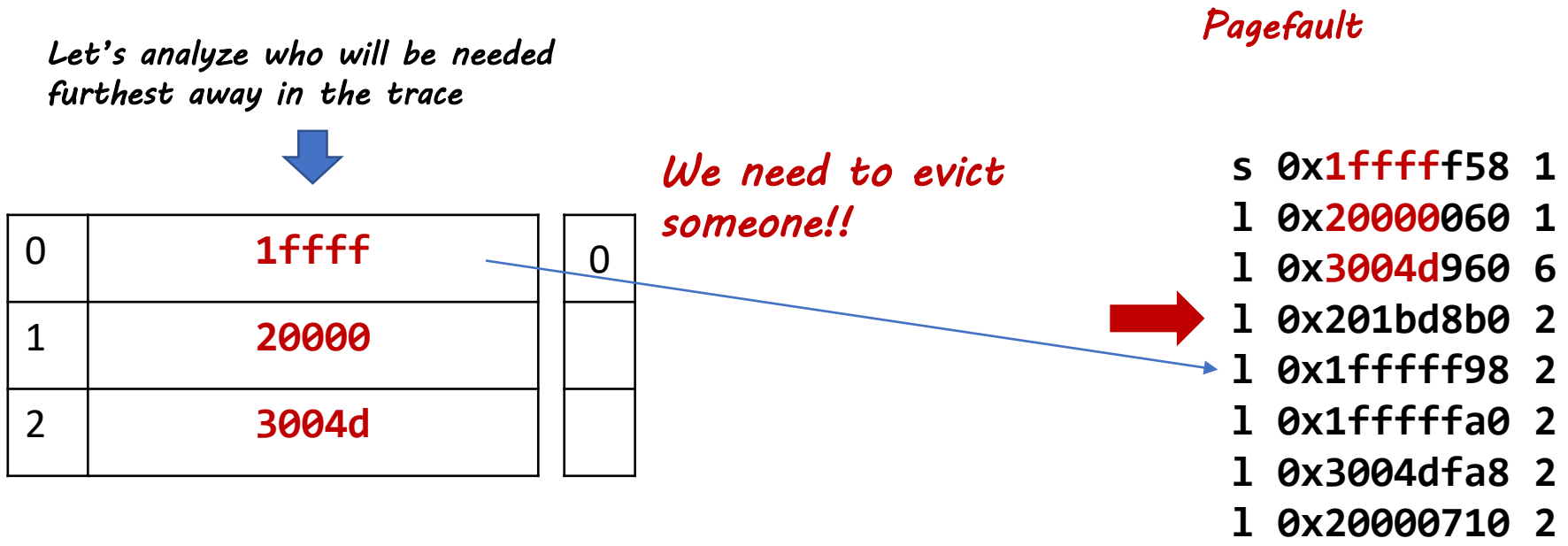
Project 3 – Optimal algorithm

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



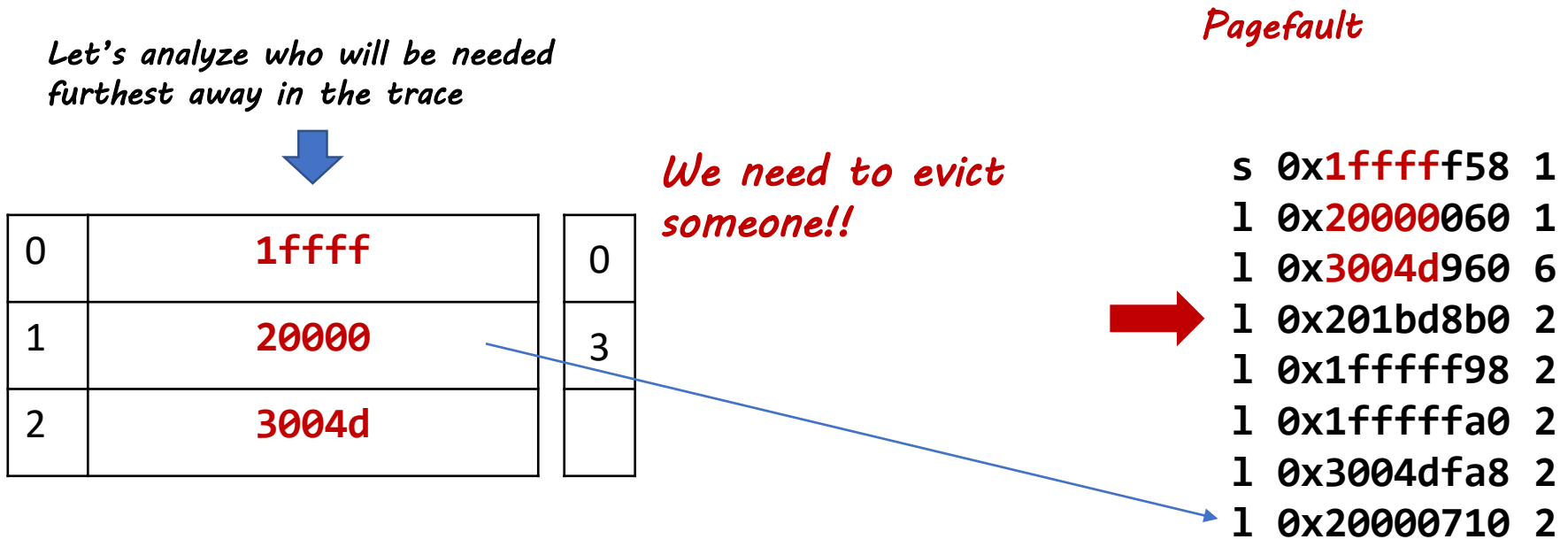
Project 3 – Optimal algorithm

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



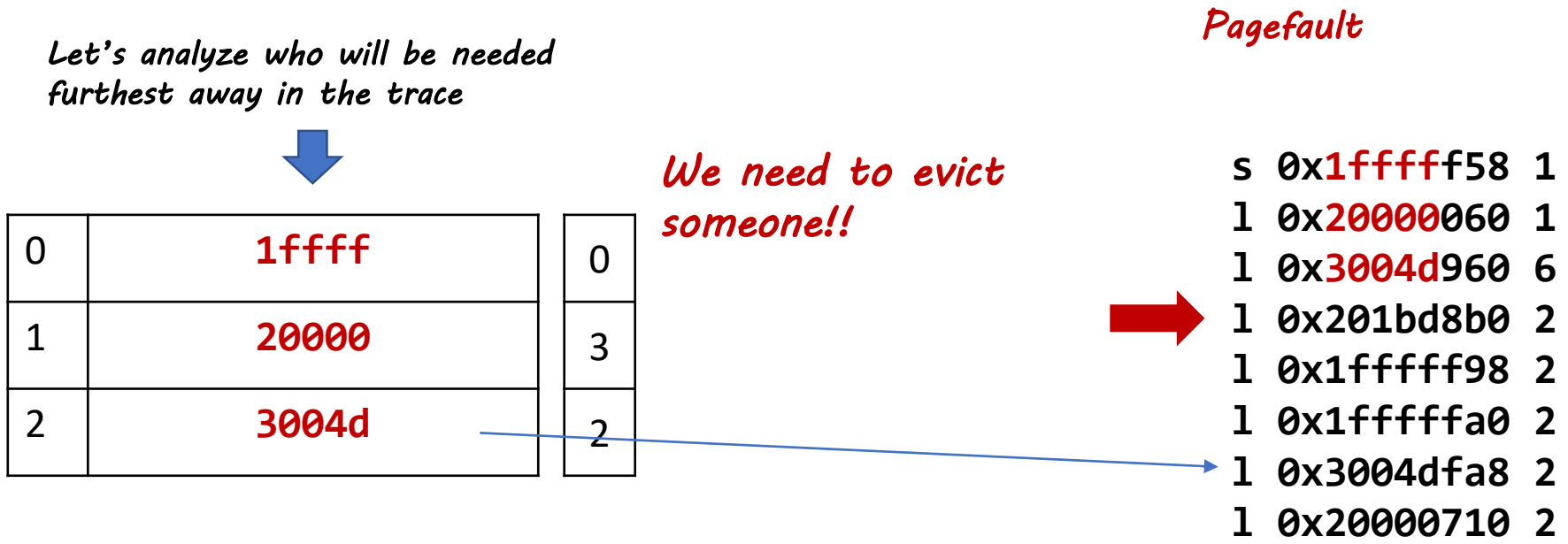
Project 3 – Optimal algorithm

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



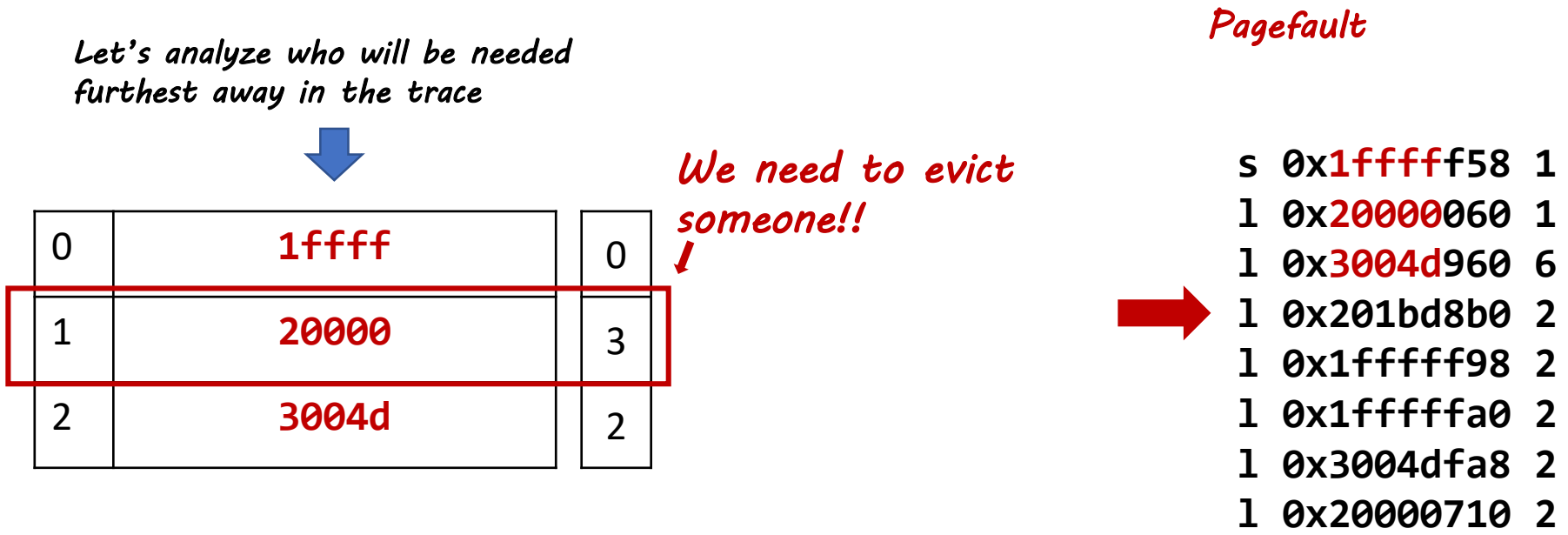
Project 3 – Optimal algorithm

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



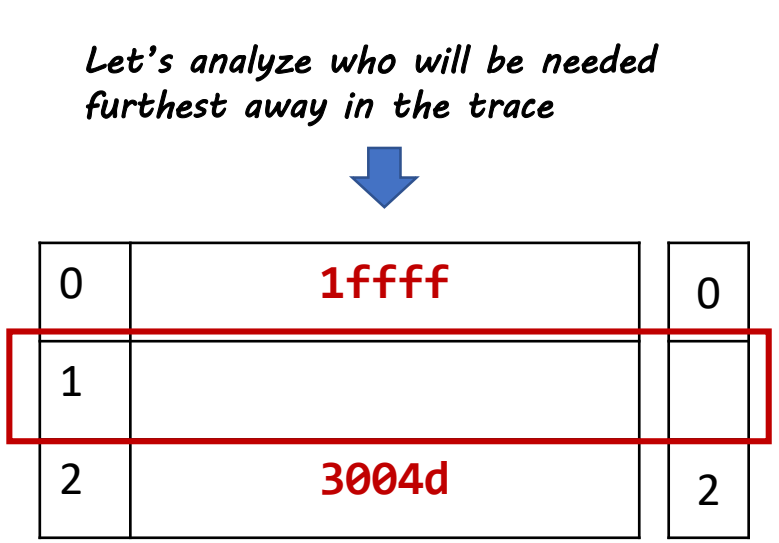
Project 3 – Optimal algorithm

- 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.

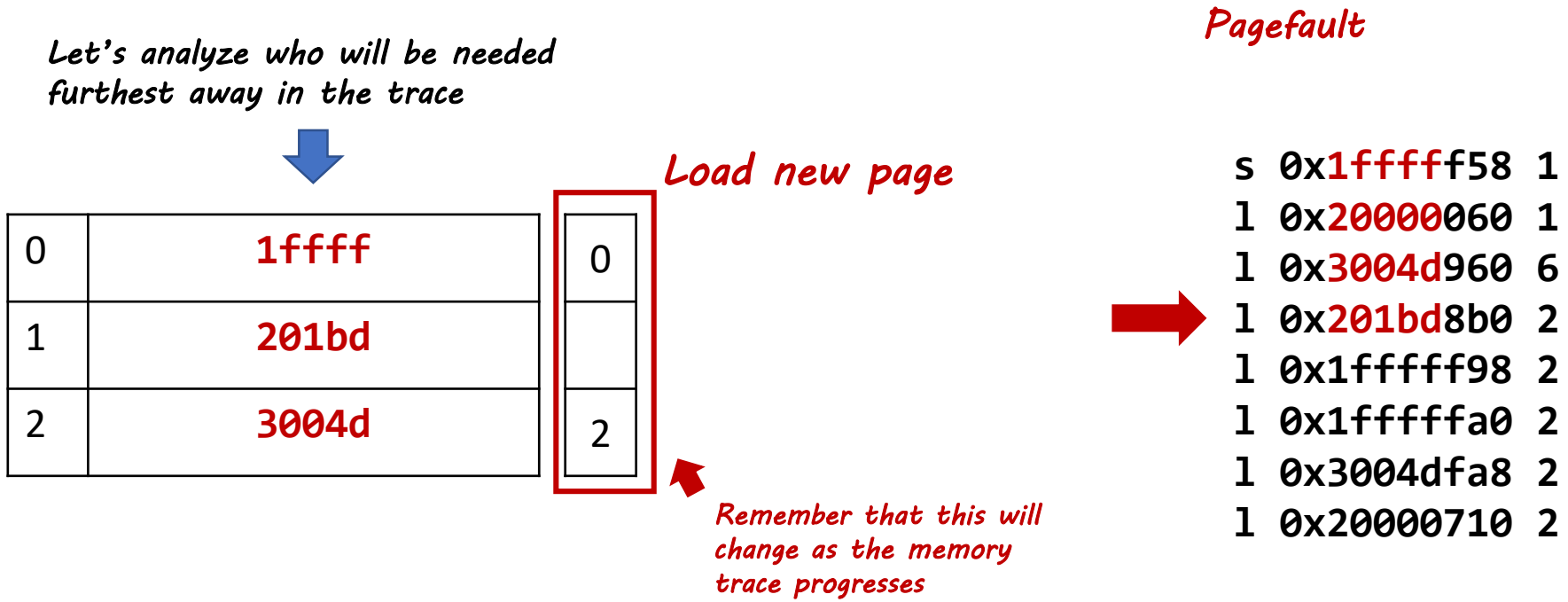


Pagefault

```
s 0x1fffff58 1
l 0x20000060 1
l 0x3004d960 6
l 0x201bd8b0 2
l 0x1fffff98 2
l 0x1ffffa0 2
l 0x3004dfa8 2
l 0x20000710 2
```

Project 3 – Optimal algorithm

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



Requirement: It should **not** take more than **5 minutes** to run your program

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

	Referenced	Counter(8 bits)	Page address
0			
1			
2			

Assume refresh parameter is 10

```
s 0x1fffff58  0
l 0x20000060  3
l 0x1fffff98  1
l 0x201bd8b0 10
l 0x3004dfa8 11
```

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

	Referenced	Counter(8 bits)	Page address
0	0	10000000	**1ffff**
1			
2			

New loaded page

Assume refresh parameter is 10

```
➡ s 0x1fffff58  0
  l 0x20000060  3
  l 0x1fffff98  1
  l 0x201bd8b0 10
  l 0x3004dfa8 11
```

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

	Referenced	Counter(8 bits)	Page address
0	0	10000000	**1ffff**
1	0	10000000	**20000**
2			

Assume refresh parameter is 10

```
  s 0x1fffff58  0
➡ l 0x20000060  3
  l 0x1fffff98  1
  l 0x201bd8b0 10
  l 0x3004dfa8 11
```

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

	Referenced	Counter(8 bits)	Page address
0	1	10000000	**1ffff**
1	0	10000000	**20000**
2			

Assume refresh parameter is 10

```
  s 0x1fffff58  0
  l 0x20000060  3
➡ l 0x1fffff98  1
  l 0x201bd8b0 10
  l 0x3004dfa8 11
```

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

	Referenced	Counter(8 bits)	Page address
0	1	10000000	**1ffff**
1	0	10000000	**20000**
2			

Assume refresh parameter is 10

```
  s 0x1fffff58  0
  l 0x20000060  3
  l 0x1fffff98  1
➡ l 0x201bd8b0 10
  l 0x3004dfa8 11
```

Before this access, there are already 10 cycles passed, refresh first

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

	Referenced	Counter(8 bits)	Page address
0	0	11000000	**1ffff**
1	0	010000000	**20000**
2			

Assume refresh parameter is 10

```
  s 0x1fffff58  0
  l 0x20000060  3
  l 0x1fffff98  1
➡ l 0x201bd8b0 10
  l 0x3004dfa8 11
```

Before this access, there are already 10 cycles passed, refresh first

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

	Referenced	Counter(8 bits)	Page address
0	0	11000000	**1ffff**
1	0	01000000	**20000**
2	0	10000000	**201bd**

Assume refresh parameter is 10

```
  s 0x1fffff58  0
  l 0x20000060  3
  l 0x1fffff98  1
➡ l 0x201bd8b0 10
  l 0x3004dfa8 11
```

Before this access, there are already 10 cycles passed, refresh first

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

	Referenced	Counter(8 bits)	Page address
0	0	11000000	**1ffff**
1	0	01000000	**20000**
2	0	10000000	**201bd**

Assume refresh parameter is 10

```
  s 0x1fffff58  0
  l 0x20000060  3
  l 0x1fffff98  1
  l 0x201bd8b0 10
➡ l 0x3004dfa8 11
```

Another 10 cycles passed,
refresh first

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

	Referenced	Counter(8 bits)	Page address
0	0	01100000	**1ffff**
1	0	00100000	**20000**
2	0	01000000	**201bd**

Assume refresh parameter is 10

```
  s 0x1fffff58  0
  l 0x20000060  3
  l 0x1fffff98  1
  l 0x201bd8b0 10
➡ l 0x3004dfa8 11
```

Another 10 cycles passed,
refresh first

Project 3 – Aging

- Evicts pages that has the lowest counter value.
  - Periodically reduce the counter (right shift by 1 bit)
  - On reference, set leftmost bit of a counter

Assume refresh parameter is 10

	Referenced	Counter(8 bits)	Page address
0	0	01100000	**1ffff**
1	0	10000000	**3004d**
2	0	01000000	**201bd**

New load

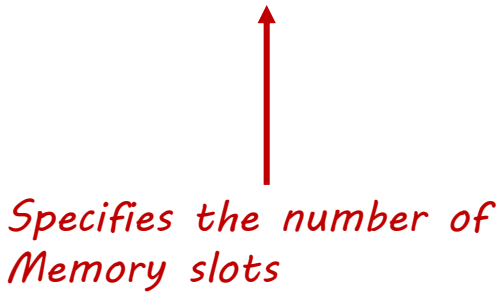
```
  s 0x1fffff58  0
  l 0x20000060  3
  l 0x1fffff98  1
  l 0x201bd8b0 10
➡ l 0x3004dfa8 11
```

*In this project, break the tie based on the **dirty flag** then based on the virtual page number (evict **the lowest numbered page**). (Real-world: random)*

Project 3 – Program interface

- Program UI

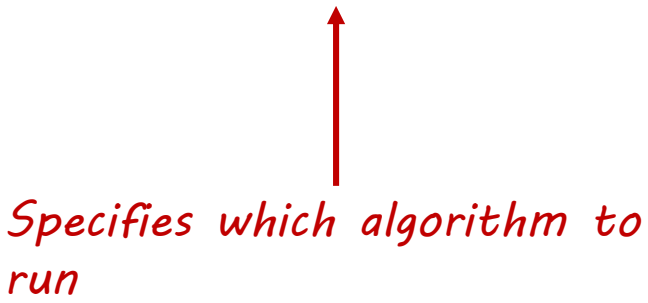
```
./vmsim -n <numframes> -a <opt|aging|fifo> [-r <refresh>] <tracefile>
```



Project 3 – Program interface

- Program UI

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



Project 3 – Program interface

- Program UI

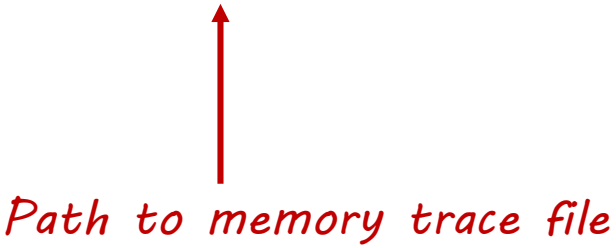
```
./vmsim –n <numframes> -a <opt|aging|fifo> [-r <refresh>] <tracefile>
```

*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>
```



CS 1550 – Project 3

- **Due**: Friday, November 08, 2019 @11:59pm
- **Late**: Sunday, November 10, 2019 @11:59pm
  - 10% reduction per late day