

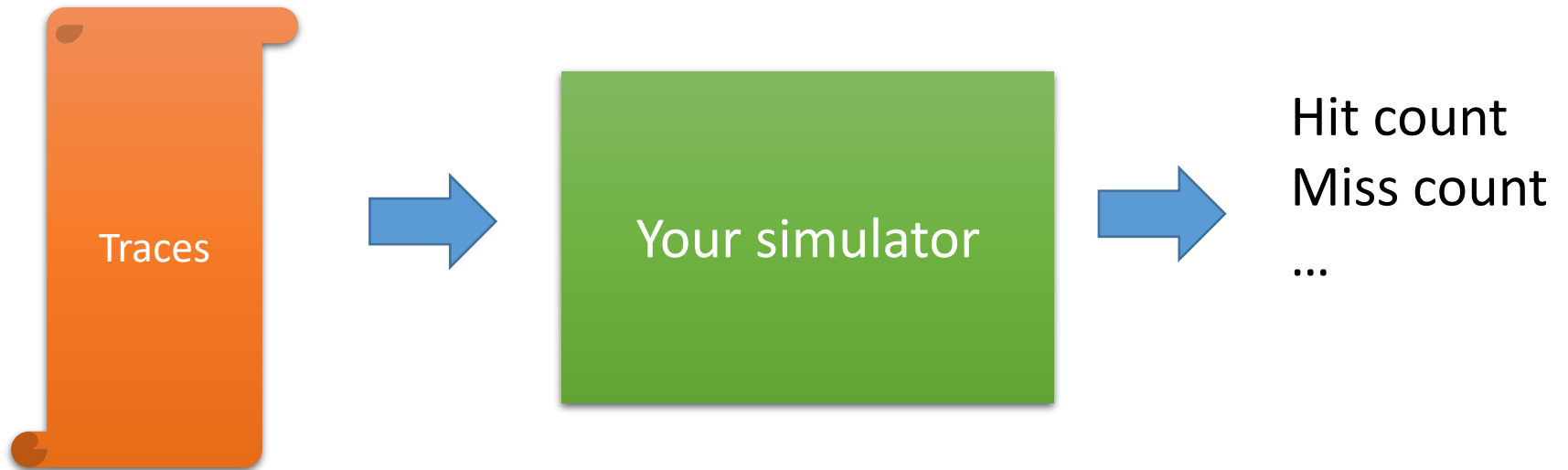
Operating Systems

Programming Assignment #5

Page Replacement Simulation: LRU and LFU

Prof. Li-Pin Chang
CS@NYCU

Simulation



Trace File Format

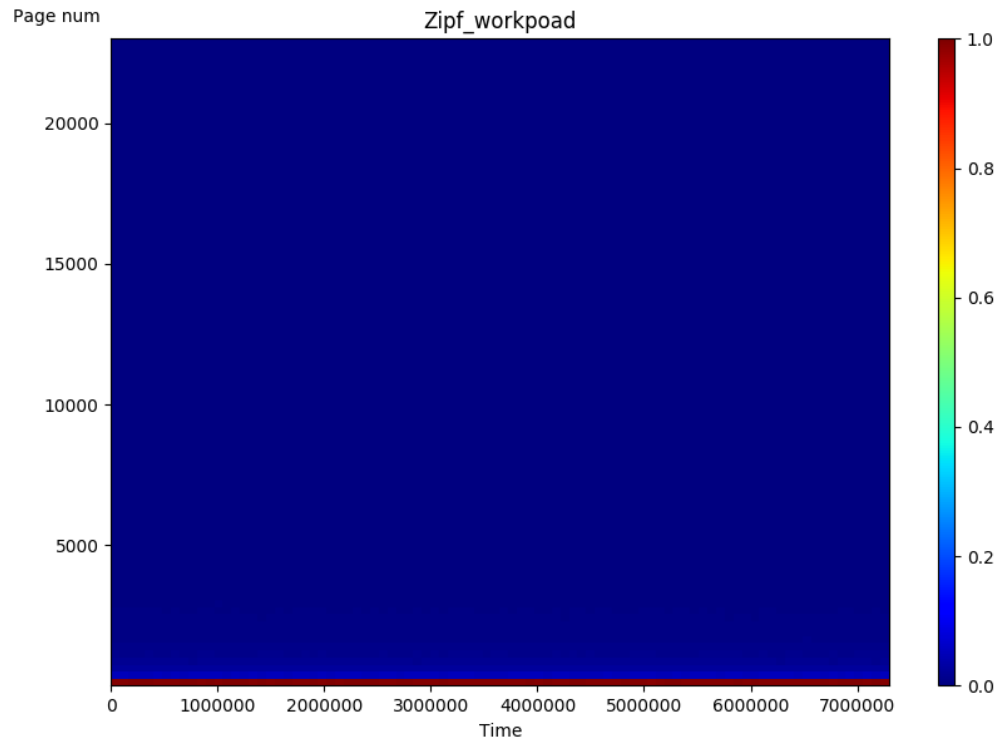
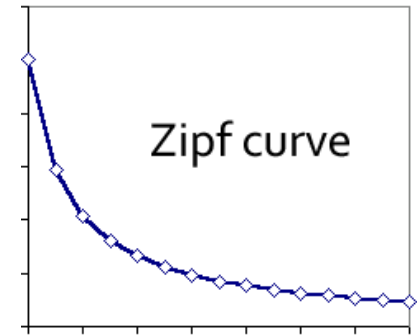
- Page number of referenced pages

```
1003  
1003  
9340  
1243  
1108  
1786  
1066  
1312  
1000  
1000  
1213  
1249  
2116
```

Unsigned integer

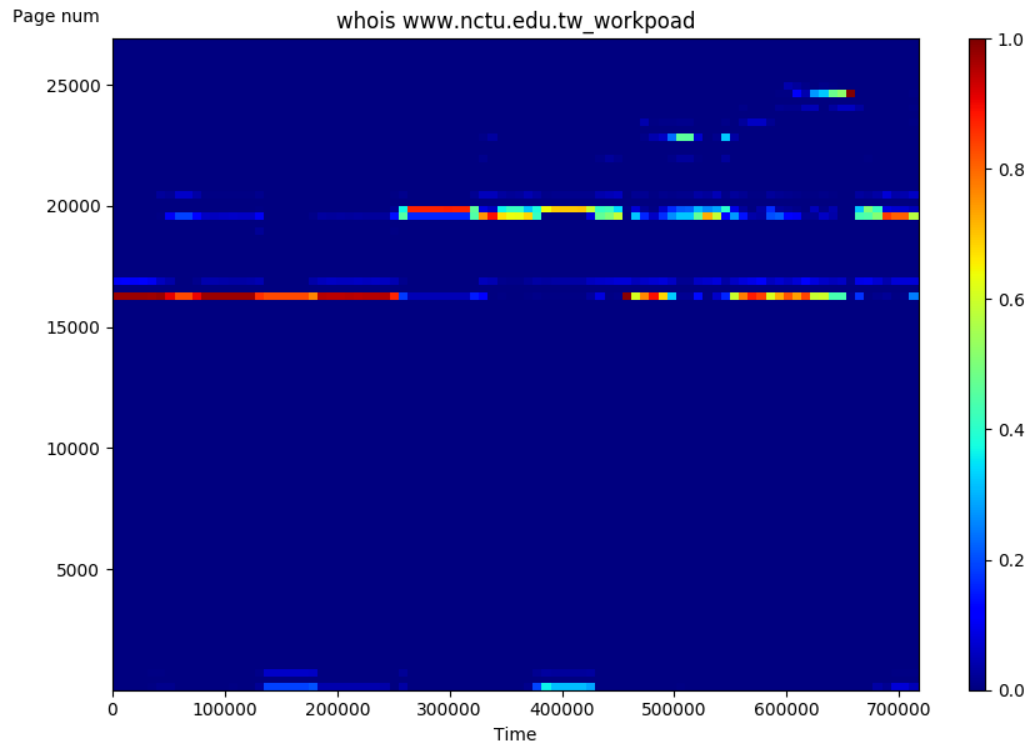
Test Workload (I)

- Zipfian distribution



Test Workload (II)

- “whois www.nctu.edu.tw”



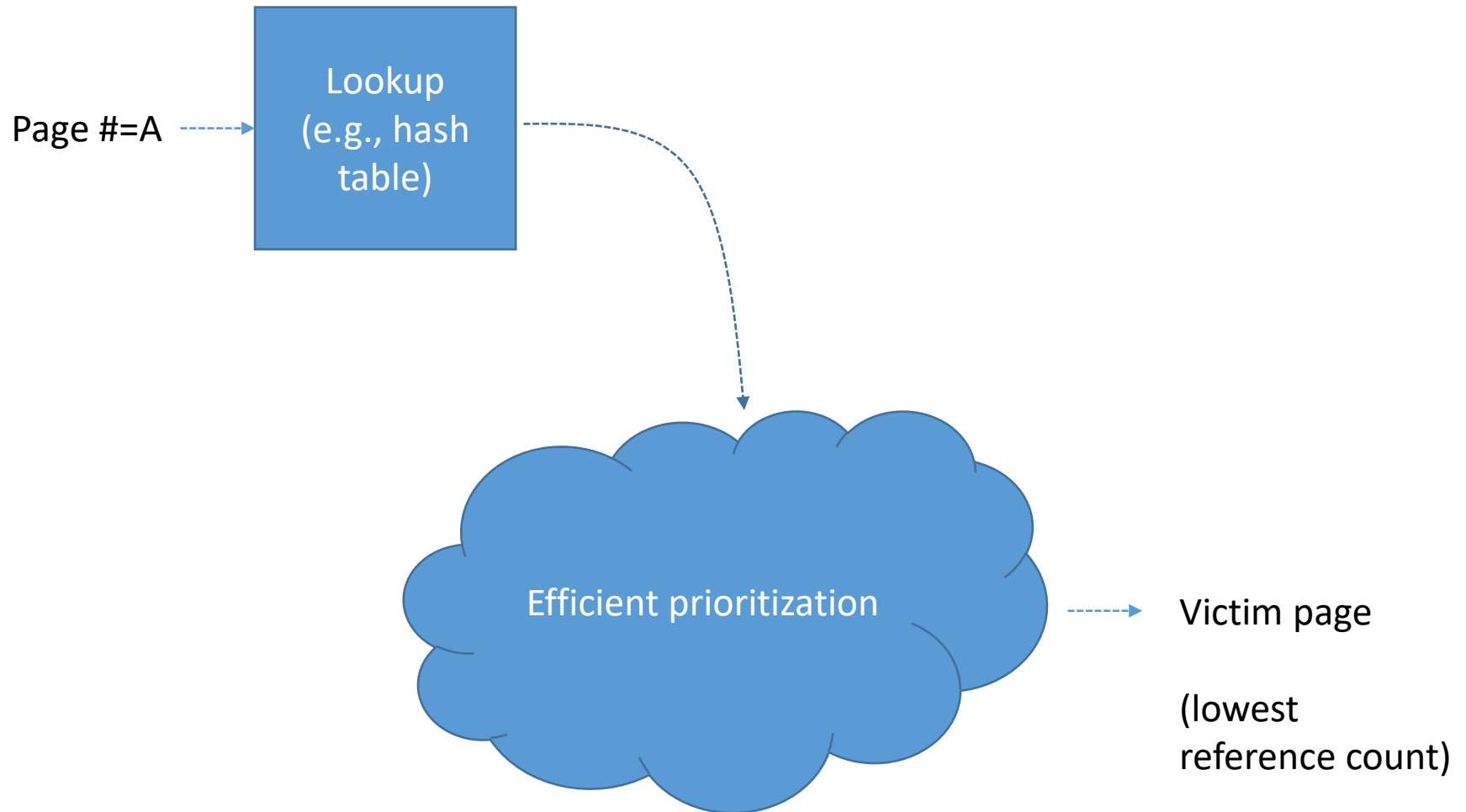
Page Replacement(LFU)

- Example: Frame #=2

	LFU	Ref count			
	4000	4001	4002		
4001 (miss)	4001	0	1	0	
4001 (hit)	4001	0	2	0	
4000 (miss)	4000	4001	1	2	0
4002 (miss)	4000 ← 4002	4001	0	2	1
4000 (miss)	4002 ← 4000	4001	1	2	0

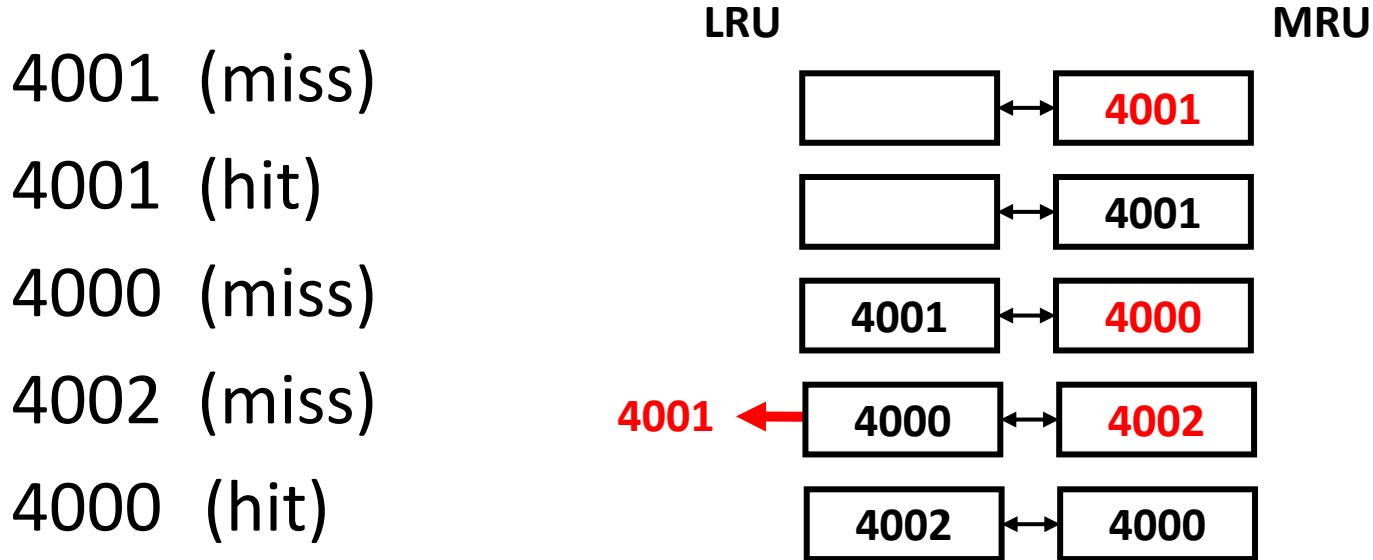
Placement candidates are those in-memory pages

Simulator Structure (LFU)



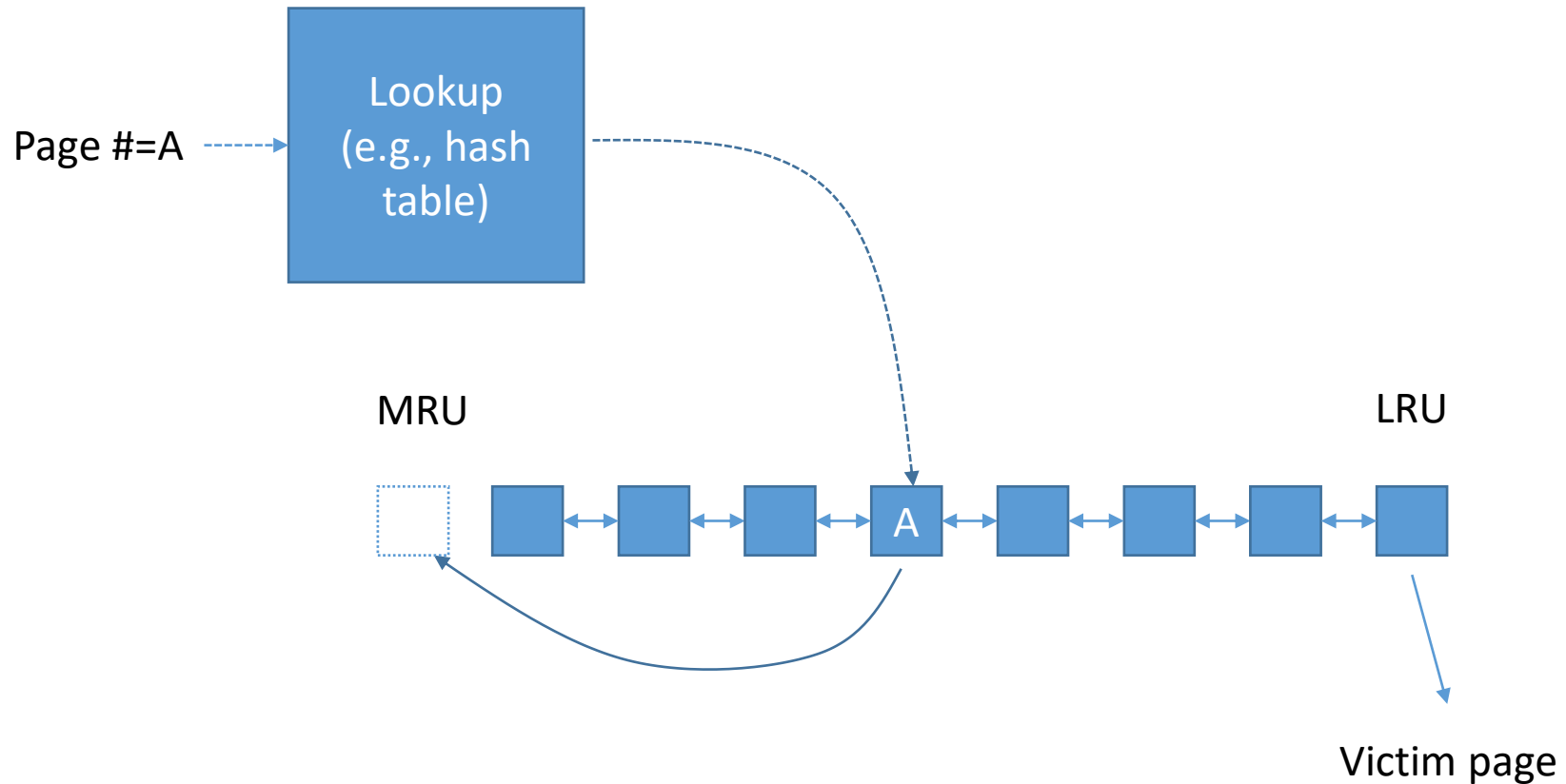
Page Replacement(LRU)

- Example: Frame #=2



Placement candidates are those in-memory pages

Simulator Structure (LRU)



Page Cache Operations

- Page lookup
 - Check whether or a new reference is a hit or a miss
 - Hash tables, binary search trees, skip lists....
- Do not use linear search!!!
 - You will receive a grade penalty if you do
 - Implement your own search, or reuse any existing libraries/classes for searching
 - TAs will examine your code
 - Duplication in this part does not count

Victim selection

- LFU
 - The least frequently used page
 - If two pages have the same access count, the page having a smaller reference sequence number is replaced
 - You may need to store the reference sequence number when a page is added to the page cache
- LRU
 - The least recently used page

Procedure

1. Algorithm=LFU
2. For (frame # = 64, 128, 256, and 512) do
 - Read the trace file
 - Run simulation
 - Print out the hit count, miss count, page fault ratio
3. Print out the total elapsed time of Step 2
4. Algorithm=LRU
5. For (frame # = 64, 128, 256, and 512) do
 - Read the trace file
 - Run simulation
 - Print out the hit count, miss count, page fault ratio
6. Print out the total elapsed time of Step 5

The scenario of program

Output format

LFU policy: (\n)

Frame	(\t)	Hit	(\t\t)	Miss	(\t\t)	Page fault ratio(\n)
64	(\t)	%d	(\t\t)	%d	(\t\t)	%.10f(\n)
128	(\t)	%d	(\t\t)	%d	(\t\t)	%.10f(\n)
256	(\t)	%d	(\t\t)	%d	(\t\t)	%.10f(\n)
512	(\t)	%d	(\t\t)	%d	(\t\t)	%.10f(\n)

Total elapsed time %.4f sec(\n) (\n)

LRU policy: (\n)

Frame	(\t)	Hit	(\t\t)	Miss	(\t\t)	Page fault ratio(\n)
64	(\t)	%d	(\t\t)	%d	(\t\t)	%.10f(\n)
128	(\t)	%d	(\t\t)	%d	(\t\t)	%.10f(\n)
256	(\t)	%d	(\t\t)	%d	(\t\t)	%.10f(\n)
512	(\t)	%d	(\t\t)	%d	(\t\t)	%.10f(\n)

Total elapsed time %.4f sec(\n)

The scenario of program

- Your output should be exactly the same as follows
- Test case : whois www.nctu.edu.tw

```
ubuntu@ubuntu:~/C-project/OS/oshw5$ ./osSummerHW5 ./testcase/whois.txt
LFU policy:
Frame   Hit           Miss           Page fault ratio
64      689115         83143         0.1076622062
128     729104         43154         0.0558802887
256     771124         1134          0.0014684212
512     771921         337           0.0004363827
Total elapsed time 0.1785 sec

LRU policy:
Frame   Hit           Miss           Page fault ratio
64      771056         1202          0.0015564746
128     771663         595           0.0007704679
256     771909         349           0.0004519215
512     771921         337           0.0004363827
Total elapsed time 0.1916 sec
```

- Your output should be exactly the same as follows
- Test case : zipfian

```
ubuntu@ubuntu:~/C-project/OS/oshw5$ ./osSummerHW5 ./testcase/zipf.txt
LFU policy:
Frame    Hit           Miss           Page fault ratio
64       7880135         2119865         0.2119865000
128      8382340         1617660         0.1617660000
256      8807903         1192097         0.1192097000
512      9134227         865773          0.0865773000
Total elapsed time 2.5885 sec

LRU policy:
Frame    Hit           Miss           Page fault ratio
64       7215568         2784432         0.2784432000
128      7880805         2119195         0.2119195000
256      8425224         1574776         0.1574776000
512      8871063         1128937         0.1128937000
Total elapsed time 2.0990 sec
```

Performance Reference

- AMD Ryzen 5 5600X 6-Core (3.70 GHz)

```
eugenechouy /mnt/d/TA_0S/hw5 ./LFU zipf.txt
64      7880135 2119865 0.2119865000
128     8382340 1617660 0.1617660000
256     8807903 1192097 0.1192097000
512     9134227 865773 0.0865773000
Elapsed time: 5.0876 sec
eugenechouy /mnt/d/TA_0S/hw5 ./LRU zipf.txt
64      7215568 2784432 0.2784432000
128     7880805 2119195 0.2119195000
256     8425224 1574776 0.1574776000
512     8871063 1128937 0.1128937000
Elapsed time: 1.944260 sec
```

- CS Workstation

```
[lwshen@linux4 ~/abc]$ ./LFU zipf.txt
64      7880135 2119865 0.2119865000
128     8382340 1617660 0.1617660000
256     8807903 1192097 0.1192097000
512     9134227 865773 0.0865773000
Elapsed time: 12.0425 sec
[lwshen@linux4 ~/abc]$ ./LRU zipf.txt
64      7215568 2784432 0.2784432000
128     7880805 2119195 0.2119195000
256     8425224 1574776 0.1574776000
512     8871063 1128937 0.1128937000
Elapsed time: 4.050614 sec
```

Performance tip:

Do not use `map/unordered_map` from STL as they are poorly implemented and very slow.

Correctness

- The TAs will prepare **another workload** to validate your implementation
- Except the total elapsed time, your results should be exactly the same as ours
- Do not use linear search in anywhere of your program; otherwise, you will receive a score penalty

More details

- Total request # ≤ 10 millions
- Highest page address == 0xfffff
- The path+file name of the trace file is an argument of your program (see the screen shot), do not hard-coding the pathname of the trace file
- For each iteration, you should open the file, run the simulation, print the result and close the file
- Do not store the trace data in memory to speed up the next iteration
- Use `gettimeofday()` to get the total elapsed time

Header of your .c or .cpp

```
/*
```

```
Student No.: <your student id>
```

```
Student Name: <your name>
```

```
Email: <your email>
```

```
SE tag: xnxcctxuxoxsx
```

```
Statement: I am fully aware that this program is not  
supposed to be posted to a public server, such as a  
public GitHub repository or a public web page.
```

```
*/
```

Testing OS Environment

- Ubuntu 18.04
- Install as a VM or on a physical machine

Credits

- 呂柏勳 helped design this project
- Questions should be directed to TAs, **not** Mr. 呂