CS 130 Operating Systems 1
Homework Assignment #2
(Due at 11:59pm Dec. 13, 2020)

- Please type the solutions using WORD, LaTex, etc, or write by hand very neatly and legibly, comparable to typing
- Please pay special attention to the DUE DATE – no late turn in or special case consideration
- Please submit your homework as a PDF file to Gradescope
- Please work on your homework individually
- The file name should be in a form of "id-YourName-hw2", for example "12345-ZhangSan-hw2.pdf"

1. [15 points] **Replacement Policy**
Consider the following page reference string:

a, b, c, d, a, b, d, c, c, b, a

With 3 frames, how many page faults would occur with the following page replacement algorithms? Fill in the tables accordingly.

Hint: all frames are initially empty, so your first unique pages will all cost one fault each.

1.1. FIFO

Page	A	B	C	D	A	B	D	C	B	A
1										
2										
3										

How many misses will you get with FIFO?

1.2. LRU

Page	A	B	C	D	A	B	D	C	B	A
1										
2										
3										

How many misses will you get with LRU?

1.3. MIN

Page	A	B	C	D	A	B	D	C	B	A
1										
2										
3										

How many misses will you get with MIN?

2. [10 points] **Address Translation**
Consider a machine with a physical memory of 8 GB, a page size of 8 KB, and a page table entry size of 4 bytes. How many levels of page tables would be required to map a 46-bit virtual address space if every page table fits into a single page?

3. [15 points] **Demand Paging**
An up-and-coming big data startup has just hired you do help design their new memory system for a byte-addressable system. Suppose the virtual and physical memory address space is 32 bits with a 4KB page size.

Suppose you know that there will only be 4 processes running at the same time, each with a Resident Set Size (RSS) of 512MB and a working set size of 256KB. What is the minimum amount of TLB entries that your system would need to support to be able to map/cache the working set size for one process? What happens if you have more entries? What about less?

4. [20 points] **Inverted Page Tables**
Consider the following case:
- 64-bit virtual address space
- 4 KB page size
- 512 MB physical memory

4.1 How much space (memory) needed for a single level page table?

Hint1: How many entries are there?
1 per virtual page.

Hint2:What is the size of a page table entry?
access control bits + physical page #.

4.2 Linear Inverted Page Table
What is the size of the hashtable? What is the runtime of finding a particular entry?
Assume the following:
- 16 bits for process ID
- 52 bit virtual page number (same as calculated above)
- 12 bits of access information

5. [40 points] **File System**
5.1 (1)Calculate the maximum file size for a file in FAT.
(2)Calculate the maximum file size for a file in the Unix file system(FFS).
(Assume a block size of 4KiB. In FAT, assume file sizes are encoded as 4 bytes. In FFS block pointers are 4 bytes long.)

5.2 (1) What are the advantages of an inode-based file system design compared to FAT?
(2) Why do we have direct blocks? Why not just have indirect blocks?
(3) Consider a file system with 2048 byte blocks and 32-bit disk and file block pointers. Each file has 12 direct pointers, a singly-indirect pointer, a doubly-indirect pointer, and a triply-indirect pointer. How large of a disk can this file system support?

5.3 Rather than writing updated files to disk immediately, many UNIX systems use a delayed write- behind policy in which dirty disk blocks are flushed to disk once every x seconds. List two advantages and one disadvantage of such a scheme.