EC 440
Spring 2021
Final
05/06/21 8PM - 05/07/21 8PM
Time Limit: 120 Minutes

Name (Print): _____

BUID _____

This exam contains 13 pages (including two cover pages) and 13 problems. Enter all requested information on the top of this page.

You are required to show your work on each problem on this exam. The following rules apply:

- **Organize your work**, in a reasonably neat and coherent way, in the space provided. Work scattered all over the page without a clear ordering will receive very little credit.
- **Mysterious or unsupported answers will not receive full credit**. A correct answer, unsupported by explanation, will receive no credit; an incorrect answer supported by substantially correct calculations and explanations might still receive partial credit.
- For multiple choice questions, put a checkmark beside every correct answer, **many** of the questions have **multiple** correct answers. Points will be deducted both for every incorrect answer checked and for every correct answer not checked.

Problem	Points
1	41
2	10
3	6
4	6
5	6
6	8
7	10
8	6
9	7
10	6
11	15
12	6
13	6
Total:	133

This is an open book test. You have two hours to finish it from time of download on gradescope. While you can take the exam at any two hour window in the twenty four hour exam period, instructors will be available between 8PM-10PM EDT on the 6th and between, 8AM-12PM and 6-8PM on the 7th, and as much as possible at other times through the exam period. If you have questions, and an instructor is available, they will be on Zoom at the regular course link:

https://bostonu.zoom.us/j/91404923667?pwd=YmJ4T0tiakMvYmtBU2hieTNocUVyQT09

What you're allowed to do: Since you are allowed to access external materials for this exam, we are not requiring a lockdown browser. You may Google for publicly available information if you wish, even if you only have one device available to you; however be aware of the time limit for the exam. In addition we are allowing you to access all questions through the entire exam; you can decide in what order you wish to complete questions, and can double-check and revise your work before submission.

What you're not allowed to do: You are expected to adhere to Boston University's academic code. Although we are allowing you to access external materials and information during the exam, you are required to perform all the work required to solve the exam problems, and to do that work during the time allotted.

Honor code

By taking this exam, I agree that:

1. I am the person who is supposed to be taking this exam and no one is helping me (physically or electronically)
2. I will not copy the exam questions in part or whole by any means
3. I will refrain from discussing this exam with anyone other than my professor until after May 8, 2021
4. I will adhere to the academic code outlined in the Student Handbook

1. Multiple choice questions. Put a checkmark beside every correct answer, **many** of the questions have **multiple** correct answers. Points will be deducted both for every incorrect answer checked and for every correct answer not checked.

   (a) (3 points) What is the difference between kernel mode and user mode in the x86 architecture?
   - ◯ While all instructions are available in both modes, and have the same logical behavior, some will cause a segment fault in user mode.
   - ◯ You can only modify the page table pointer in kernel mode.
   - ◯ All operating system functionality in a micro-kernel operating system runs in kernel mode.

   (b) (4 points) Typically what is most memory used for by the OS?
   - ◯ To maintain the data and text of programs that are currently running.
   - ◯ For buffering network packets.
   - ◯ To maintain the data and text of programs that have recently run.
   - ◯ For caching file system data.

   (c) (3 points) What is a system call?
   - ◯ A way for an application to make a request to the kernel.
   - ◯ A way for devices to tell the operating system that a device operation has completed.
   - ◯ A mechanism to transfer control between user mode and kernel mode.

   (d) (3 points) Dynamic analysis tools are useful to programmers to:
   - ◯ Ensure that there are no syntax errors in the code.
   - ◯ Identify sources of memory leaks
   - ◯ Ensure that function implementations match the prototypes defined in a header.

   (e) (3 points) In a system with only fixed 16KB pages
   - ◯ One can suffer from internal fragmentation.
   - ◯ One can suffer from external fragmentation.
   - ◯ One can suffer from both external and internal fragmentation.

   (f) (3 points) Contrasting user and kernel level implementations of threads.
   - ◯ User level implementations enable faster switching between threads.
   - ◯ User level implementations make it easier to exploit multiple cores.
   - ◯ You can't really combine both kinds of implementations together, where there are Z user level threads multiplexed on top of Y kernel level threads.

   (g) (7 points) Considering the different synchronization primitives we discussed in class:
   - ◯ Monitors result in better performance than the alternatives.
   - ◯ RCU can always be used to replace reader writer locks
   - ◯ Embedding locks in data that they protect generally results in better cache performance.
   - ◯ Generally mutexes result in easier to understand code than monitors.
   - ◯ MCS locks result in better scalability than regular spin locks because many readers can lock the data simultaneously.
   - ◯ NUMA can result in starvation in implementation of spin locks.
   - ◯ Linux uses two phased locking to prevent deadlock.

(h) (3 points) In the ext family of file systems, what is the purpose of Block Groups?

- ◯ To maintain cylinder locality, where directories, inodes and blocks in the file are co-located in the same portion of the disk.
- ◯ To avoid replication of blocks across the entire disk.
- ◯ To avoid disk fragmentation; maintaining large extents of free blocks on the disk.

(i) (3 points) What is the major source of bugs in operating systems?

- ◯ Memory management because it is the most complex part of the operating systems.
- ◯ Device drivers because there are so many devices.
- ◯ Schedulers and File systems because there are so many different options.

(j) (3 points) Log structured file systems were introduced:

- ◯ To avoid the complexity of soft updates for maintaining meta-data consistency in the file system.
- ◯ Under the assumption that writes will dominate I/O to disk as reads are buffered in the file cache.
- ◯ Because seeks make random access slower than sequential disk I/O.

(k) (3 points) What is the advantage of an inverted page table over a multi-level page table?

- ◯ Reverse the decision so the page translation is obvious.
- ◯ Avoids multiple dependent load operations traversing through multiple page tables.
- ◯ Better support for large pages.

(l) (3 points) DMA from a device:

- ◯ Avoids having the CPU involved in every data transfer from the device.
- ◯ Avoids data going over the memory bus twice.
- ◯ Avoids security threat since the CPU is in charge of all copying.

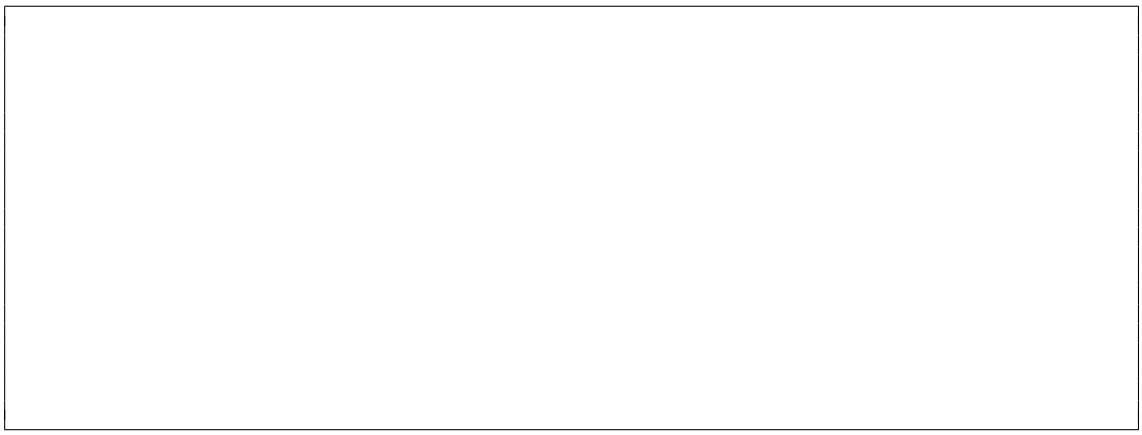
2. Consider the following code that implements a non-thread-safe pushdown stack:

```
struct XX {
  struct XX *next;
  int contents;
};

struct XX_stk {
  struct XX *items;
};

void XX_stk_push(struct XX_stk *s, struct XX *item) {
  item->next = s->items;
  s->items = item;
}
struct XX *XX_stk_pop(struct XX_stk *s) {
  struct XX *val = s->items;
  if (val != NULL)
    s->items = val->next;
  return val;
}

int XX_stk_empty(struct XX_stk *s) {
  return s->items == NULL;
}
```

(a) (5 points) Make this structure thread-safe, so that multiple threads may call `_push/_pop/_empty` simultaneously without crashing. Do this by adding: a) a mutex as part of the `XX_stk` structure b) calls to `lock/unlock` that mutex within the `_push` and `_pop` calls. Use an arrow and text on the code to the right to indicate where to add mutex and lock/unlock calls

(b) (5 points) What is wrong with the following code? How can it crash with a null pointer (i.e. segmentation fault) when run by multiple threads at the same time?

```
while (!XX_stk_empty(s)) {
  struct XX *item = XX_stk_pop(s);
  printf("%d\n", item->contents);
}
```



3. (6 points) What are the two main purposes of an operating system?

4. (6 points) Describe what the VFS layer of an operating system does and why operating systems have this layer.

5. Suppose that the virtual page reference stream contains repetitions of long sequences of page references followed occasionally by a random page reference. For example, the reference stream: 0, 1, ... , 511, **431**, 0, 1, ... , 511, **332**, 0, 1, ... consists of repetitions of the sequence 0, 1, ... , 511 followed by a random reference to pages 431 and 332.

   (a) (3 points) Why will the standard replacement algorithms (LRU, FIFO, clock) not be effective in handling this workload for a page cache that is smaller in size than what would be needed to contain the entire 0, 1, ..., 511, ### sequence?

   (b) (3 points) If this program were allocated 500 page frames, describe a page replacement approach that would perform much better than the LRU, FIFO, or clock algorithms.

6. (8 points) A computer with a 57-bit address uses a four-level page table, where 11 bits are used to index into each page table. How large are the pages and how many are there in the address space?

7. Disk read requests come in for disk blocks on cylinders 10, 5, 39, 6, and 100 in that order. A seek takes 4 msec per cylinder. Assume the arm is currently at cylinder 27 and was last at cylinder 22.

   For each of the following algorithms, write out the order that disk requests will be serviced, and then the total seek time that will be consumed.

   For example, if requests where handled in the order they arrived, the order will be:

   ```
   10-5-39-6-100
   ```

   And the total seek time is $(17 + 5 + 34 + 33 + 94) * 4msec = 734msec$

   (a) (6 points) Elevator Algorithm:

   (b) (3 points (bonus)) First-Come First-Served:

(c) (3 points (bonus)) Shortest Seek First:

(d) (4 points) Which algorithm is normally used with hard drives and why?

8. (6 points) What is the purpose of randomizing the contents of the process address space?

9. (7 points) What type of multiplexing (time, space, or both) is used for memory pages and CPU cores? Please explain.

10. (6 points) What is the difference between tail latency and average latency, and why has tail latency become more important in the cloud?

11. Five jobs, A through D, arrive at a computer at almost the same time. They have estimated running times shown in the table below. All jobs are completely CPU bound. The higher the number the higher the priority.

Job	Run time	Priority
A	10	5
B	2	2
C	5	1
D	3	4

Write the order that the jobs will complete in and the average time processes took to run.

(a) (5 points) Shortest job first

(b) (5 points) Priority scheduling

(c) (5 points) First-come, first-served, assuming that jobs arrived in the order A through D.

(d) (10 points (bonus)) For round robin scheduling, assuming that each job gets an equal share of the CPU when running and ignore process switching overhead.

12. (6 points) Name three of the interesting things that virtualization has been used for.

13. (6 points) Describe three key reasons why many people use the cloud.