Name: _____ **CompID:** _____

For each question, either (a) mark one circle or (b) circle one of T or F for each row. Blank questions will be weighted slightly higher than incorrectly marked questions.

Feel free to write clarifying comments next to any question or option that needs them.

. . . . . .

Question 1 Assume that `x` is a `size_t` representing a virtual address. Let `P` be the number of page-offset bits and `L` be the number of levels of page table. Assume every page table entry take up 8 bytes.
Write a C expression for the first virtual page number used from `x` in address translation. You may define extra variables to help organize your answer if you wish.

Answer: _____

Question 2 Compare concurrency and parallelism

- ○ Things can be concurrent, or parallel, or both, or neither
- ○ Nothing is both concurrent and parallel
- ○ Concurrent and parallel are synonyms
- ○ Anything that is parallel is also concurrent, but not vice-versa
- ○ Anything that is concurrent is also parallel, but not vice-versa

Information for Q3–Q6 Immediately after `fork`ing a new process, the new process has…

Question 3 Operating system data (PID, file descriptors, permissions, etc)

- ○ is shared with the parent process (change one process's and the other's changes too)
- ○ has the same values as the parent process, but can be independently updated
- ○ has (at least some) distinct values from the parent process

Question 4 PC, condition codes, etc. that

- ○ have the same values as the parent process, but can be independently updated
- ○ have (at least some) distinct values from the parent process
- ○ are shared with the parent process (change one process's and the other's changes too)

Question 5 Program registers that

- ○ have the same values as the parent process, but can be independently updated
- ○ have (at least some) distinct values from the parent process
- ○ are shared with the parent process (change one process's and the other's changes too)

Question 6 User-mode memory (virtual address space, value stored at each address, etc)

- ○ is shared with the parent process (change one process's and the other's changes too)
- ○ has the same values as the parent process, but can be independently updated
- ○ has (at least some) distinct values from the parent process
- ○ some regions of memory are one of the above, others are another

Information for Q7–Q10 Immediately after `pthread_create`ing a new thread, the new thread has…

Question 7 Operating system data (PID, file descriptors, permissions, etc)

- ○ is shared with the parent thread (change one thread's and the other's changes too)
- ○ has the same values as the parent thread, but can be independently updated
- ○ has (at least some) distinct values from the parent thread

Question 8 PC, condition codes, etc. that

- ○ have the same values as the parent thread, but can be independently updated
- ○ have (at least some) distinct values from the parent thread
- ○ are shared with the parent thread (change one thread's and the other's changes too)

Question 9 Program registers that

- ○ have the same values as the parent thread, but can be independently updated
- ○ have (at least some) distinct values from the parent thread
- ○ are shared with the parent thread (change one thread's and the other's changes too)

Question 10 User-mode memory (virtual address space, value stored at each address, etc)

- ○ is shared with the parent thread (change one thread's and the other's changes too)
- ○ has the same values as the parent thread, but can be independently updated
- ○ has (at least some) distinct values from the parent thread
- ○ some regions of memory are one of the above, others are another

Question 11 In the code below, `dothis` is executed by several threads, each with a different multiple-of-1000 argument. Add `pthreads_...` calls to blank lines as needed to remove any race conditions.
Assume all globals are properly initialized and that `isprime` uses only local variables. Your solution should minimize thread wait time.

```
pthread_mutex_t m1, m2, m3;
pthread_barrier_t b1, b2, b3;
int *primes;
int num_primes;
int primes_capacity;

void *dothis(void *arg) {
    int start = (int)arg;


    for(int i = start; i < start + 1000; i += 1) {


        if (isprime(i)) {


            while (num_primes == primes_capacity) {


                primes_capacity *= 2;


                primes = realloc(primes, sizeof(int)*primes_capacity);


            }


            primes[num_primes++] = i;


        }


    }


    return NULL;
}
```

Information for Q12–Q13 In the following, "each X is twice as large" means "has twice as many of the things that an X stores." For example, if X was a set of arrays of strings, this would double the number of arrays per set, not the number of strings per array.

For each, pick the most correct answer. If needed, assume the pre-modified cache is a 32KB 2-way set-associative cache with an eight-bit set index.

Question 12 If I modify a cache to have half as many blocks, but each block is twice as large, the modified cache will be

- ○ faster for programs exhibiting more temporal than spatial locality
- ○ faster for programs exhibiting more spatial than temporal locality
- ○ faster for most programs
- ○ none of the above

Question 13 If I modify a cache to have half as many sets, but each set is twice as large, the modified cache will be

- ○ faster for programs exhibiting more temporal than spatial locality
- ○ faster for programs exhibiting more spatial than temporal locality
- ○ faster for most programs
- ○ none of the above

Information for Q14–Q16 Let $H(z)$ compute a cryptographic **h**ash of z, $U(z)$ apply your p**u**blic key to z, $R(z)$ apply your p**r**ivate key to z, $S(z)$ apply a symmetric key to z, and $C(y, z)$ concatenate y and z.

Question 14 A cryptographically signed version of document x is computed as

Answer: _____

Question 15 Diffie-Helman is used to enable computing

- ○ S
- ○ R
- ○ H
- ○ C

Question 16 Which one the following is true?

- ○ if you can compute $U(x)$, you can also compute its inverse $U^{-1}(x)$
- ○ if you can compute $S(x)$, you can also compute its inverse $S^{-1}(x)$
- ○ if you can compute $R(x)$, you can also compute its inverse $R^{-1}(x)$
- ○ if you can compute $H(x)$, you can also compute its inverse $H^{-1}(x)$

Question 17 Suppose a memory access traverses the page table down to the last layer before creating a page fault, and the page fault causes the OS to generate a segfault signal which your program handles. TLB entries for which of the following might have been **added** to the TLB during this sequence of events?

- T F user signal handler code pages
- T F user call stack pages
- T F the data page the memory access attempted to read
- T F kernel page-fault handler code pages
- T F kernel call stack pages
- T F intermediate page table pages for the data page causing the fault

Question 18 Suppose a single-byte memory access to address `0x10000` adds an entry to an otherwise empty 32KB 2-way set-associative cache with an eight-bit set index. If the next instruction is also a single-byte memory access, what is the largest address that would result in a cache hit?
Show your work so we can assess your approach, not just your solution.

Answer: _____

Question 19 When using the HTTPS protocol, TLS is used to secure communication between a client (such as a web browser) and a server. This means third parties are prevented from learning which of the following?

T F IP address of the server

T F IP address of the client

T F Contents of request sent from client to server

T F Contents of reply sent from server to client

. . . . . .

Pledge On my honor as a student, I have neither given nor received assistance on this exam.

{Signature}