NYU, Tandon School of Engineering

Bridge to Computer Science Program

4th Exam

Thursday 17 December 2020

- You have two hours
- There are 100 points total.
- Note that there are longer problems at the end. Be sure to allow enough time for these.
- We supplied you with a file, named 'solutions.txt', where you should type all your answers.
- Write your name, netID and NYU ID at the head of this file.
- For editing this file, you are allowed to use any compiler including CLion, XCode and Visual Studio.
- Calculators are not allowed.
- This is a closed-book exam. No additional resourced are allowed.
- Pay special attention to the style of your code. Indent your code correctly, choose meaningful names for your variables, define constants where needed, choose most suitable control statements, etc.
- In all questions you may assume that the users enter inputs as they are asked. For example, if the program expects a positive integer, you may assume that users will enter positive integers.
- No need to document your code in this exam, but you may add comments if you think they are needed for clarity.
- Read every question completely before answering it.
- **When done, please upload your answer file to Newclasses.nyu.edu, Gradescope and email to dkatz@nyu.edu**

1) (3 pts)On a system with only one CPU, when a process calls “fork,” the result will be that one process will be in the running state and the other will be in which state?
    a. New
    b. Ready
    c. Blocked
    d. Exit

2) (3 pts) A section of code with must run “atomically,” or without interruption would be consider to be a?
    a. Thread
    b. Peterson’s algorithm
    c. Critical section
    d. Mutual Exclusion

3) (3 pts) The processor “mode” is determined by a single bit inside
    a. Main memory
    b. The process’s PCB
    c. The system bus
    d. The registers
4) (3 pts)A page which is no longer in main memory will have it’s _____ bit set to zero.

5) (3 pts) Transmitting voice and video over a network usually uses the _____ layer 4 protocol.

6) (10 pts) In a system that supports kernel level threads, each thread will have it’s own stack pointer. Explain where the stack pointer will be stored and why each thread needs its own stack.

7) (10 points) Explain how circuit-switched networks and packet-switched networks differ. Provide a “real-world” example of when you would use each of them.

8) (10 pts) Assuming our domain has three authoritative DNS servers, one master and two slaves, explain what would happen if the master went offline for a long period of time.

9) (10 pts) When using TCP for data transfer, the sender would expect to see an ACK for all data transmitted. However, after transmitting 100KB of data, the sender receives back three ACKs for 50KB of data. Explain why this is and what action the sender should take.

10) (15 pts) Your company is assigned an IP subnet of 10.1.0.0/22. You are asked to divide that into three networks. Two of the networks, New York and Los Angeles, will have 150 devices each. The third network is a Point-To-Point WAN link between the NY and LA network. Provide the network number and subnet mask (or CIDR mask) for each of the three networks.

11) (10 pts) A router's forwarding table has the following entries. Given a destination address of 192.168.1.32, which next hop IP address will be used? Explain how you came to that answer.

Network Number	Next Hop	Metric
0.0.0.0/0	10.10.1.2	25
192.168.0.0/22	10.10.1.5	10
192.168.0.0/23	10.10.1.6	20

12) (20 pts) We are designing a program to hold information in a calendar. The structure of a Calendar is simple enough, it's a linked list of calendar appointments and one function to get the total duration of all appointments. However, calendar appointments are a bit more complicated. Some calendar appointments are one-time appointments (i.e. Exam 4, Thursday 17 December 2020 @ 6PM) and some are recurring (i.e. Weekly Webinar, every Thursday @6PM).

All calendar appointments have a start and end time. A common way to record times is with an integer number of seconds (Unix timestamp where the epoch is midnight, January 1st 1970) so we will use this technique. Allow for the construction of SingleCalendarEntry objects and RepeatingCalendarEntry objects with the start and end times being passed in as two integers. Additionally, RepeatingCalendarEntry objects will have a number of times that they repeat. All calendar entry objects have a function called "duration" which returns the number of seconds that the appointment will last. For SingleCalendarEntry objects, this is just endTime-startTime. However, for RepeatingCalendarEntry objects this is: ((endTime-startTime)*numberOfTimesRepeating).

Your task is to design the classes necessary to support the calendar and both calendarEntryObjects described above. Inside the calendar you may only store ONE linked list (use the STL list class, please). You must, also, design a function in the calendar to report the total number of seconds for which the calendar has appointments. Put simply, this is the sum of all of the durations of calendar entry objects stored in the calendar.

(n.b.: You are not being asked to write code to add or remove entries to the calendar, someone else will do that).