SER 334 Midterm Exam: Part 2 of 2

Fall 2017, October 6nd, 2017

Student #1 Last Name: BHATTACHARYYA
Student #1 First Name: DEBARATI
Student #1 Signature: Debarati Bhattacharyya

Student #2 Last Name (if applicable): Suryawanshi
Student #2 First Name (if applicable): Pratik
Student #2 Signature (if applicable): Pratik

Exam Instructions

This part of the SER334 midterm exam is open book (any), open note (any), and open internet. Optionally, you are permitted to collaborate with one other student in the course. The exam must be completed on paper and submitted in-person to the instructor or electronically via BlackBoard as a PDF of images. Write legibly and use a pen. There are 12 points available, and the exam must be completed and submitted on BlackBoard by 11:59:00PM (AZ time) on 10/6/2017. A grace period of 18 hours will also be in effect, allowing you to submit until 5:59:00PM (AZ time) on 10/7/2017. All submissions after the grace period will receive a zero score for lateness.

Test Takers

This take-home exam may be taken by an individual or a pair. The graded score for this exam will be given to both students. If only one student is taking this exam, then the four areas on this labeled "Student #2" must be filled out with the phrase "NO OTHER STUDENT". If students begin to collaborate on the exam, then they are obligated to complete the exam together.

Honor Code

Students MUST adhere to ASU's Academic Integrity policy (https://provost.asu.edu/academic-integrity/policy) throughout the process of completing this exam. Mainly: the work that you submit must assess the individual knowledge and skills of the ASU student(s) named above. Any misrepresentation of work, or collaboration with persons not named above, is grounds for integrity sanctions up to and including an X grade in the course. Some specific expectations are:

- You may not collaborate with anyone other than the student(s) named above. Collaboration includes discussing any part (e.g., questions, solutions) of this exam with any other person (excluding your instructor).
- Exam answers must demonstrate your knowledge. It is not acceptable (even if cited) to take text from another source. Rewording material does not change the fact it comes from another source.
- No material from this exam may be redistributed.

Before you begin this exam, both students MUST write the following: "This exam will demonstrate the individual knowledge and skill of the student(s) named at the top of this page. I ([*write your name here in parenthesis*]) have read the cover of this exam, the Student Obligations to Academic Integrity (from ASU's Academic Integrity policy), and will follow all rules specified." in the space below. If any student fails to state this, the exam will automatically be given a zero grade.

Student #1:
This exam will demonstrate the individual knowledge and skill of the student(s) named at the top of this page. I (Debarati Bhattacharyya) have read the cover of this exam, the Student Obligations to Academic Integri (from ASU's Academic Integrity policy) and will follow all rules specified.

Student #2 (if applicable):
This exam will demonstrate the individual knowledge and skill of th students named at top of this page. I (pratik suryawanshi) have read the cover of this exam, the student obligations to Academic integrity (from ASU's Academic Integrity policy) and will follow all rules specified.

Short Answer: C Programming [4 points]

1. [Acuña] Consider the following (incorrect) function which adds a node to the tail of a list. (Assume the list is never empty.) The struct *grade_node* is used to represent a node containing grade information in a linked list of grades. [4 points total]

```
struct grade_node {
    int value;
    char assignment[255];
    struct grade_node* next;
};

void add_last(struct node* head, struct node* node) {
  struct node new_node = *node;

  while(head->next != null)
    head = head->next;

  head->next = &new_node;
}
```

What is the issue with this code? Explain with C terminology. [2 points]

ns:- The issue with this code is that the head and the node both are local variables. The assignment of it to head will not be visible outside the scope of this function. Hence, the list will remain unchanged. Also, the variables are using the stack memory (since they are local). Hence, as soon as the function exits, the memory will be returned to the system and made available for next function call.

2. Why is it more natural to implement an ADT in an OOP language like Java than a procedural language like C? [2 points]

s:- ADT is an user-defined data type where the representation of and operations on objects of the type are defined in a syntactic unit. Procedural programming - on the other hand (like that in C) is code that is broken into procedures. These procedures manipulate the data and produce the result. In OOP, data and functions however are bundled together in object. Therefore, there is no requirement of any shared or global data in case of OOP.

Short Answer: Processes [2 points]

3. Assume that you have a set of nine computers running different processes that are all collaborating to solve a problem. In terms of reducing total system cost, would it be better to make the processes communicate with shared memory (centralized on a special machine) or message passing? Explain. [2 points]

: It would be better to make the processes communicate via shared memory. This is because, in case of message passing, there will be more system calls involved apart from read, write. Hence, it will incur more overhead which will in turn increase the overall system cost.

Short Answer: Threads [2 points]

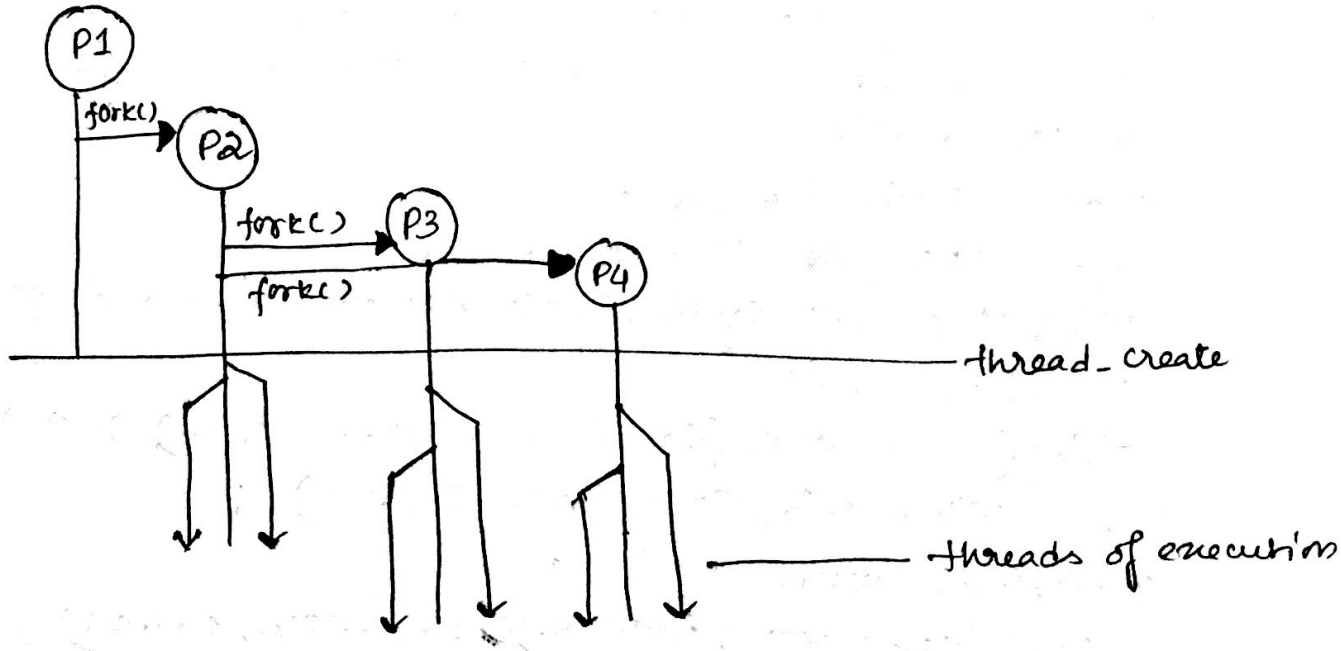
4. [Silberschatz 4.15 edited] Consider the following code fragment and answer the following questions:

```
pid_t pid, pid2;
pid = fork();

if (pid == 0) {
    pid2 = fork();
    if(pid != 0)
        pid = fork();
    thread_create(...);
    thread_create(...);
}
```

Using "lifeline notation", draw the creation of processes and threads during execution. [2 points]

Ans:-



Scenario: Operating-System Structures [4 points]

5. In any computer task, a series of system calls are executed utilizing the kernel's API. Imagine that you are using a VOIP program like Skype to talk with someone - what would be some system calls you imagine for network access, sound, and screen display? Use psuedo-code to show what you think the program's main loop is doing. Be sure that the functions you use are worthy of being a system call.

Ans:- RECEIVER SIDE :-

```
// socket function for socket connection.
   connection_var = socket(domain, type, protocol)
// connection port opened.
 if(connect(connection_var, SOCKETADDRESS, ADDRESSLENGTH)==-1)
     connection error message;
 Loop- continuously listens to and receives data.
 // read the data from server.
 info = read(connection_var, BUFFER, BUFFERSIZE, FLAGS)
 Now, segregate the received data as sound and image.
 SOUND-BUFFER = sound , DISP-BUFFER = image
 If sound data -
 // open audio device file
   fd = open(/dev/file, O_WRONLY)
 Using the ioctl system call set the attributes for the device file.
 // write to audio
 while(val = write(fd, SOUND-BUFFER, SIZE) > 0)
       write(fd, SOUND-BUFFER, val);
// write to screen
  write(conn_var, DISP-BUFFER, SIZE, FLAGS).
```

SENDER SIDE:-

```
// socket function for socket connection
    connection_var = socket(domain, type, protocol)
// connection port opened
  if(connect(connection_var, SOCKETADDRESS, ADDRESSLENGTH)==-1)
     connection error message;
// read from device file and store in buffer (AUDIO).
   read(device file pointer, BUFFER, SIZE).
```

// Read from device file and store in buffer (CAMERA/DISPLAY)
read (device file pointer, BUFFER, SIZE).

Based on the common timestamp, form the packet and write the data onto the connection
write (connection_var, data buffer, sizeof(data buffer)).

// Close the connection
close (connection_var).