

Calculators may be used in this examination provided they are not capable of being used to store alphabetical information other than hexadecimal numbers

UNIVERSITY OF BIRMINGHAM

School of Computer Science

Operating Systems and Systems Programming

Main Summer Examinations 2024

Time allowed: 2 hours

[Answer all questions]

Note

Answer ALL questions. Each question will be marked out of 20. The paper will be marked out of 80, which will be rescaled to a mark out of 100.

Question 1

- (a) What is a pointer in C? How can it be used to write a function which returns several values? **[4 marks]**
- (b) Give two possible consequences of an access beyond the boundaries of an array in a C program. **[4 marks]**
- (c) Consider the following code:

```

1  #define BUFFERLENGTH 80
2  struct entry_t {
3      int entry_id;
4      char title[BUFFERLENGTH];
5  };
6  struct list_t {
7      struct entry_t entry;
8      struct list_t *next;
9  };
10
11 struct list_t *addElem(struct list_t *allItems, int newID,
12     char *newTitle) {
13     struct list_t *prevItem, *currentItem;
14     struct list_t newItem;
15
16     strcpy(newItem.entry.title, newTitle);
17     newItem.entry.entry_id = newID;
18     currentItem = allItems;
19     while(currentItem->entry.entry_id < newID) {
20         prevItem = currentItem;
21     }
22     prevItem->next = &newItem;
23     return allItems;
24 }
25 void deleteAllItems (struct list_t *items) {
26     while(items != NULL) {
27         free(items);
28         items = items->next;
29     }
30 }
```

The intention is that the function `addElem` has a list `allItems`, a title ID `newID` and a title `newTitle` as arguments and adds a new element consisting of the ID and the title to the list. You may assume that the list is ordered by the title ID when the function is called. You may assume that `newTitle` points to a properly null-terminated string of arbitrary length. The list which is returned by `addElem` should also be ordered by title ID.

The function `deleteAllItems` should delete the list and all titles stored in the list. This code compiles correctly, but does not work as intended. List the errors (including memory management) and provide corrections. Do not list or provide corrections for concurrency errors.

Hint: The `strcpy(char *dest, const char *src)` function copies the string pointed to by `src`, including the terminating null byte (`'\0'`), to the buffer pointed to by `dest`. It does not do any memory allocation or checks. The function `strlen` returns the length of a string excluding the terminating null byte (`'\0'`). **[12 marks]**

Question 2

(a) What is a critical section?

[4 marks]

(b) Why is it so difficult to find synchronisation errors?

[4 marks]

(c) Consider the following code fragment:

```

1  #define MEALS_AVAILABLE 80
2  #define SEATS_AVAILABLE 100
3  struct seat_t {
4      bool available;
5      bool meal_booked;
6  };
7  int noOfMealsAvailable = MEALS_AVAILABLE;
8  struct seat_t allSeats[SEATS_AVAILABLE];
9
10 int reserveSeatMeal(bool wantsMeal) {
11     for (int i = 0; i < SEATS_AVAILABLE; i++) {
12         if (allSeats[i].available) {
13             if (wantsMeal) {
14                 if (noOfMealsAvailable > 0) {
15                     allSeats[i].available = false;
16                     allSeats[i].meal_booked = true;
17                     noOfMealsAvailable --;
18                     return i;
19                 }
20             }
21             else { /* doesn't want meal */
22                 allSeats[i].available = false;
23                 allSeats[i].meal_booked = false;
24                 return i;
25             }
26         }
27     }
28     return -1;
29 }
30
31 void listAllBookings () {
32     for (int i = 0; i < SEATS_AVAILABLE; i++) {
33         if (!allSeats[i].available)
34             printf ("Seat %d booked\n", i);
35         if (allSeats[i].meal_booked)
36             printf ("Meal for seat %d booked\n", i);
37     }
38 }

```

The intention is that the array `allSeats` stores which seats are reserved and whether a meal has been booked. The function `reserveSeatMeal` tries to reserve a seat and book a meal if a meal is wanted. If a meal is wanted but no meal is available a seat is not reserved. This function returns the index of the reserved seat or -1 if no seat is available. The function `listAllBookings` lists all bookings which are made.

This code is part of a multi-threaded server. This code is not thread-safe, i.e errors might happen when some of these functions are called at the same time within different threads. Identify those errors and correct them. For locks it is sufficient to indicate which kind of lock or locks you would use, and where you would lock and unlock. Your solution should maximise the degree of possible parallelism. **[12 marks]**

Question 3

- (a) During context-switching, the operating system replaces the currently executing process in the CPU by a new one. Why is it important for the operating system to avoid too many context switches and let a process run for sufficient time before another context switch? **[4 marks]**
- (b) Which steps are performed by the operating system during context-switching? **[4 marks]**
- (c) What is the primary difference between paging and segmentation? **[4 marks]**
- (d) A virtual machine allows a guest operating system to access the computer system resources through the host operating system. Draw a layered diagram that shows how applications running in both the guest and host operating systems can be run on top of the underlying hardware. Also depict the role of the virtualisation layer. You should label the host operating system to be Windows and the guest operating system to be Linux. **[8 marks]**

Question 4

- (a) The degree of multi-programming determines the number of processes to be loaded in memory and run simultaneously. The operating system schedules one of these memory-resident processes at a time on the CPU. Consider the case where the memory is full after allocating various processes in it. To allow a new process to be executed, one of the currently resident processes must be suspended (swapped out). Describe two different ways/criteria that you can utilise for selection of a process to be swapped out considering that we want to maximise CPU utilisation and minimise the response time for the user. **[4 marks]**
- (b) A computer system has four page frames. A process makes the following list of page references: 7, 1, 0, 3, 0, 2, 6, 4, 5, 3, 6, 2, 4, 2, 6, 1, 7, 0, 1. Assume that this is the only executing process, use the FIFO (First-In First-Out) page replacement algorithm to illustrate the page replacement process in the following frames. Draw the updated status of the frames in your answer book each time a page is replaced in a frame. The first four page requests have been done for you. **[10 marks]**

7											
1											
0											
3											

- (c) Consider the following three processes and their execution traces. You need to schedule them on a single processor. The scheduler uses the first-come-first-served scheduling algorithm. Three processes arrive in the order of Process1, followed by Process2, followed by Process3.

Process1: Needs CPU for 4ms and then blocks for 6ms. After that it again needs CPU for 6ms and then it terminates.

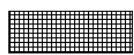
Process2: Needs CPU for 2ms and then blocks for 12ms. It then again executes for 3ms and exits.

Process3: Needs CPU for 10ms and then exits.

Using separate execution of each process, identify in the following time line when a process may be in the running state, when in the blocked and when in the ready state. Please note that the CPU is never idle and must be running one of these processes at any time until all processes have finished.

Use the following patterns to show states in the blank area in the time line.

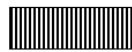
[6 marks]



Running



Ready



Blocked

0	4	8	12	16	20	24	28
---	---	---	----	----	----	----	----

Process1

0	4	8	12	16	20	24	28
---	---	---	----	----	----	----	----

Process2

0	4	8	12	16	20	24	28
---	---	---	----	----	----	----	----

Process3

Do not complete the attendance slip, fill in the front of the answer book or turn over the question paper until you are told to do so

Important Reminders

- Coats/outwear should be placed in the designated area.
- Unauthorised materials (e.g. notes or Tippex) must be placed in the designated area.
- Check that you do not have any unauthorised materials with you (e.g. in your pockets, pencil case).
- Mobile phones and smart watches must be switched off and placed in the designated area or under your desk. They must not be left on your person or in your pockets.
- You are not permitted to use a mobile phone as a clock. If you have difficulty seeing a clock, please alert an Invigilator.
- You are not permitted to have writing on your hand, arm or other body part.
- Check that you do not have writing on your hand, arm or other body part – if you do, you must inform an Invigilator immediately
- Alert an Invigilator immediately if you find any unauthorised item upon you during the examination.

Any students found with non-permitted items upon their person during the examination, or who fail to comply with Examination rules may be subject to Student Conduct procedures.