Quiz 02 Questions

Bookmark this page

Ques1

4.0/4.0 points (graded)

Consider the following pseudo code snippet:

```
static string department="CSE";
static int counter = 0;
public string getGrade(int id, String name){
    String[] grades ={"3.75", "4.00", "3.20", "3.90"};
    string result = name + " " + grades[id];
    return result;
}
```

Now, after loading the program into memory, **Identify** at which section of the process memory, the variables of the above code snippet will reside -

The variable *department* will reside in :

Data ✔

The variable *id* will reside in :

Stack ✔

The variable *grades* will reside in :

Stack ✔

The variable *result* will reside in :

Stack ✔

Submit You have used 1 of 1 attempt

Show Answer

Question 2

2.0/6.0 points (graded)

#NOTE: carefully read the hints before you start to interpret the program; everything you need to know to interpret the program correctly is explained there. Then you just have to apply your intelligence.

Question:

The following pseudocode searches for a number (the value of the variable 'searchValue') in the an array (in the global variable 'searchArray') using 4 threads from two different processes. If a thread finds the searched number in the array then it prints a line in the following format:

'X': Thread No 'Y', matching index: 'Z'

Here,
'X' should be replaced by either C or P,
'Y' should be replaced by a number from 0 to 3 depending on the value of threadNo, and
'Z' should be replaced by the value of the index where the searched number is found, which can be any number from 0 to 39 as the array has 40 elements.

Exactly two lines will be printed by the program. You have to fill in the gaps(Find the values of A,B,C,D,E,F) for the output for these two lines.

Line #1) _(A)__: Thread No __(B)__, matching index: __(C)___

Line #2) _(D)__: Thread No __(E)__, matching index: __(F)___

There is no partial marking for this question. Further, you must not mix up the lines, that is, output

of the first line cannot be written in the second line.

Hints:

---

1. fork() system call creates a child process and returns the ID of the child process to its parent.

2. wait(NULL) system call lets a parent wait for a child to complete.

2. thread_create(funcName, arg) creates a new thread that executes the funcName with arg as the argument. It returns the id of the thread being created.

3. thread_join(threadId, null) is used to wait for a thread with id = threadId to complete its execution.

4. printLine(message) prints the contents of a string message on screen on a separate line.

Program:

```
int searchArray[] = {   15, 10, 11, 5, 16, 20, 12, 19, 7, 18,
                        7, 8, 12, 16, 13, 6, 4, 1, 3, 17,
                        11, 10, 16, 3, 9, 19, 18, 15, 4, 13,
                        20, 7, 13, 2, 5, 9, 14, 17, 8, 10 };
int arrayLength = 40;
char prefix = 'P';
int processNo;

int searchedValue; // Value of this variable is given later
int searchNumber(int threadNo) {

        int searchStart = threadNo * 10;
        int searchEnd = searchStart + 10;
        for (int i = searchStart; i < searchEnd; i++) {
                if (searchedValue == searchArray[i]) {
                        string message = prefix + ": Thread No " + threadNo + ", matching index: " + i + "."
                        printLine(message);
                        return i;
                }
        }
        return -1;
}

int main() {

        int pid = 0;
        pid = fork();
        if (pid == 0) {
                prefix = 'C';
                processNo = searchedValue % 2;

        } else {
                wait(NULL);
                processNo = (searchedValue + 1) % 2;
        }

        int threadNo = processNo * 2 + 1;
        thread_t threadId = thread_create(searchNumber, threadNo);
        thread_join(threadId, NULL);
        threadNo = threadNo - 1;
        searchNumber(threadNo);
        return 0;
}
```

Enter Your Answer:

Consider, **searchedValue = 11;** and, main() is called.

Value of A [C or P]

✖ Answer: C

Value of B [within 0 to 3]

2 ✔ Answer: 2

2

Value of C [within 0 to 39]

20 ✔ Answer: 20

20

Value of D [C or P]

✖ **Answer:** P

Value of E [within 0 to 3]

2 ✖ **Answer:** 0

2

Value of F [within 0 to 39]

21 ✖ **Answer:** 2

21

Submit You have used 2 of 2 attempts

Show Answer

Answers are displayed within the problem

Question 3

0.0/5.0 points (graded)

Consider the information of following processes -

Process	Arrival Time	Burst Time
P0	0	6
P1	1	5
P2	6	8
P3	8	7

Now apply the Non-Preemptive Shortest Job First(SJF) scheduling algorithm on the above given data. You need to show your work in your PDF.

Enter Avg. Turnaround Time [Enter upto two decimal points. example: 2.50]

12.75 ✖ **Answer:** 11.50

12.75

Enter Avg. Waiting Time [Enter upto two decimal points. example: 2.50, 2.00]

6.25 ✖ **Answer:** 5.00

6.25

Submit You have used 1 of 1 attempt

Show Answer

Answers are displayed within the problem

SUBMIT YOUR PDF:

Submit your pdf file in this google form: https://forms.gle/4XZzXAH6USRVcqyn6

You can save this link. Submit your pdf within 10 minutes after your exam has ended on BuX.

Previous Next