

Let's Thread

CSL301 - Operating Systems

Home Assignment - 4



Question 1

Task:

- Write a C program that divides an array of 100 integers (values 1 to 100) into 10 segments.
- Create 11 threads:
 - The first 10 threads each compute the sum of a segment of 10 elements and return the result.
 - The 11th thread collects the 10 results from the previous threads, computes their sum, and returns the final total to the main function, which prints it.
- Use a single thread function for all threads, and avoid unnecessary variables in your implementation.

Question 1

Example Output:

- Thread 1 partial sum: 55
- Thread 2 partial sum: 155
- Thread 3 partial sum: 255
- Thread 4 partial sum: 355
- Thread 5 partial sum: 455
- Thread 6 partial sum: 555
- Thread 7 partial sum: 655
- Thread 8 partial sum: 755
- Thread 9 partial sum: 855
- Thread 10 partial sum: 955
- Total sum: 5050

Syntax Tip:

```
1 void* partial_sum(void* arg) {
2     args* a = (args*)arg;
3     int sum = 0;
4     for (int i = a->start; i < a->start + SEGMENT_SIZE; ++
        i)
5         sum += a->arr[i];
6     int* result = malloc(sizeof(int));
7     *result = sum;
8     return result;
9 }
```

Give each thread a pointer to its array segment, and collect results using `pthread_join`.

Question 2

Task:

- Each thread sums its segment of an integer array and returns the partial sum.
- The final (11th) thread sums all partial sums from the first 10 threads.
- The main process waits for the final thread, then divides the total sum by the number of elements to compute the average.
- Print the computed average in the main function.

Example Output:

- Thread 1 partial sum: 55
- Thread 2 partial sum: 155
- ...
- Thread 10 partial sum: 955
- Total sum: 5050
- Average: 50.5

Question 3: Threaded Maximum in Array Segments

Task:

- Given an array of size in multiple of N and $N+1$ threads, divide the array into N distinct segments.
- Each thread finds the maximum value within its assigned segment.
- One final thread collects all maxima from the previous threads and determines the overall maximum.
- The final result is returned to the main function.

Example Output:

- Thread 1 max: 16
- Thread 2 max: 32
- ...
- Thread N max: 100
- Overall max: 100