National University of Computer and Emerging Scie

FAST School of Computing Fall-2021 Islamabad Campus

CS-5009: Advanced Operating Systems

Serial No:

Midterm Exam

Total Time: 2 Hours

Total Marks: 90

Wednesday, 03rd November, 2021

Course Instructor

Dr. Ehtesham Zahoor

_____

Signature of Invigilator

_____ _____

Student Name Roll No Section Signature

DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED

Instructions:

1. Attempt on question paper. Attempt all of them. Read the question carefully, understand the question, and then attempt it.
2. No additional sheet will be provided for rough work. You can however use the back of last page for any rough work.
3. After asked to commence the exam, please verify that you have Nine (09) different printed pages including this title page. There are 2 questions in total.
4. Use permanent ink pens only. Any part done using soft pencil will not be marked and cannot be claimed for rechecking.
5. You need to be as specific as possible and provide the answers in the space provided. Cutting, overwriting and ambiguous answers would be considered as incorrect.
6. Please make sure to properly comment your code and write clearly, failure to do so can cost you 10 marks. Code containing overwritten text and corrections cannot be claimed for rechecking. You need to write code specifically in the space given.

The paper is open book and you can use any printed helping material. Ke
clean and do not share anything. You cannot use any electronic device in

	Q-1	Q-2	Total
Marks Obtained			
Total Marks	45	45	90

You need to be as specific as possible and provide the answers in the space provid overwriting and ambiguous answers would be considered as incorrect. It is importa and cleanly provide the answer so think before you write anything.

Question 1 – Short Questions [15x3 = 45Marks]

I. We studied about the 50,000-core supercomputer built on AWS for cancer research. How d *AWS staff and administrators* help build the supercomputer?

The answer is no because as we studied that Aws staff and administration has no contribution in building the supercomputers for cancer research.

II. In reference to the Beowulf cluster that we built, what is the purpose of the command belo and what does the | in the command do?

echo "/mirror *(rw,sync)" | sudo tee -a /etc/exports

Here the output of one command will be given as input to other . The tee command is used for read and write purpose. The \mirror folder in master node will be shared with slave nodes in read-write mode.

III. In reference to the Beowulf cluster that we built, is it compulsory to add the moun information at Slave machines to the ***/etc/fstab***

Fstab is used in order to mount it on every boot. We mount the external storage so that every slave node can get access of it . So thats why use this.

IV. In reference to the Beowulf cluster that we built, what would be the implications o connectivity if we change the home directory of mpiuser to some other directory than the /mi

As we know that only mirror folder is shared so if we change the home directory the /mirror will not be accessible by ssh.

V. Consider that n processes are using bakery algorithm for synchronization. If the token rece by a process is higher than that of all other processes, in which **state,** as per the Five-state P Model, the process would be? Write one-line justification as well.

IX. We built a docker image containing a Web server. The content of the *dockerfile* are as belo
You need to discuss the purpose of both lines clearly stating what is *nginx* and *alpine* and wha
being copied from where to where?

```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

Here we are coping the web page from Local directory to nginx html folder.

X. In the code segment below (assume all includes to be present and no syntax errors) is ther
race condition and/or critical section. If so, fix it elegantly using any synchronization primitive
we have studied. You need to only write the modified part in the ***space provided***.

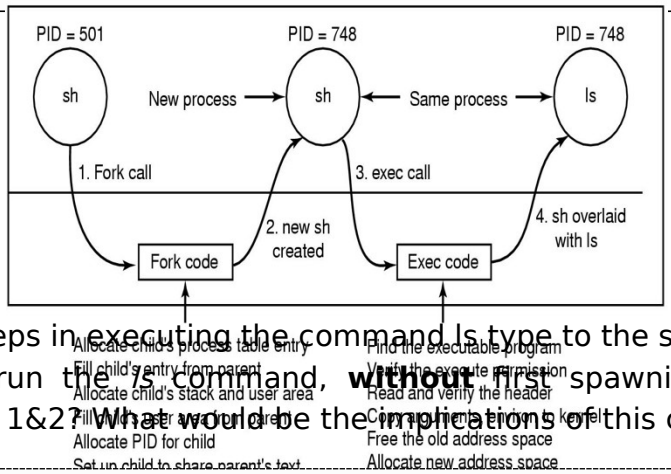
```
#define NUM_THREADS 10
int sharedData = 21;
void* incrementData(void* arg) {
    localData = sharedData;
    pthread_exit(NULL);
}
int main()
{
   pthread_t threadID;
   for (int counter=0; counter<NUM_THREADS;counter++) {
      pthread_create(&threadID, NULL, incrementData, NULL);
   }
   cout << "ThreadCount:" << sharedData <<endl;
   pthread_exit(NULL);
}
```

In the above code there is no critical section that exists.

XI. Is it possible that a computer system runs out of file descriptors? Discuss this in reference t
file_struct, a kernel-resident array data structure containing the details of open files.

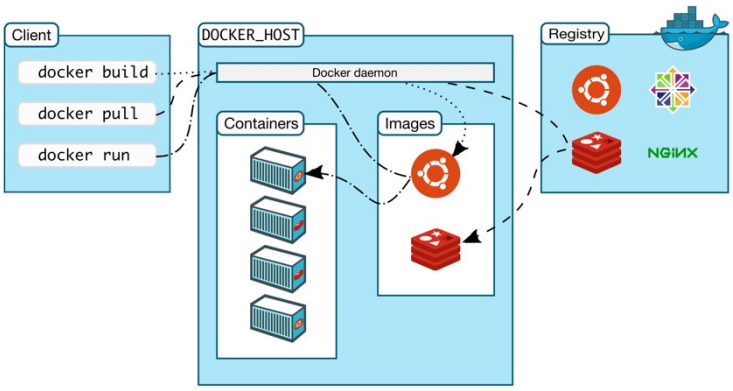
Yes it is possible that computer system runs out of file descriptors.
It contains information about the open files in it but it may end up with this.

There will be busy wait which means it will be in Ready state.



VI. Consider the steps in executing the command ls type to the shell, as shown in the figure ab
Is it possible to run the ls command, **without** first spawning a new process using fork i.e. by removing the step 1&2? What would be the implications of this choice then?

Yes it can be possible
But there will be limitation that the ls command will be executed only once and no other will be alllowed to executed after this.



VII. Consider running the hello-world docker image on your personal system for the very first ti
You need to discuss which elements act as client, docker host and the registry, show above.

Docker_hub will be registry
Our own machine will be docker host
Terminal will be client

VIII.We built a docker image containing a Web server. The command needed to run the image *`docker run -p 80:80 ehtesham/staticweb`* What is the purpose of -p flag?

It Maps the system port to the web server port. So it is basically used for port Mapping.

XII. Why everything in Unix (and its flavors) is considered as a file (or specifically a file-descript

In unix everything is considered as a file in which two functions are available read and write.

XIII. What is the logical (not syntactic) error in the code below and how it can be fixed? You nee just discuss the solution without modifying the code provided.

```
int main(int argc, char* argv[]) {
int sum, counter, inputList[6] = {11,45,3,5,12,-3};
#pragma omp parallel num_threads(2)
 {
  #pragma omp for schedule(static, 3)
   for (counter=0; counter<6; counter++) {
   sum+=inputList[counter];
   } }
printf("The summed up Value: %d", sum);}
```

Te shared variable sum is the logical error and it can be prevented by using Locks.

XIV. Why the Semaphore and pthread mutex locks are called the hardware locks?

When an instruction can do both set and lock it is hardware lock and in case of interrupts prevention hardware locking mechanism are used.

XV. How blockchain is related to our course? More specifically, Which aspect of decentr blockchain relates to our course?

It uses a support system that runs in the background of a computer system. The synchronization is related to our course.

Question 2 – Fork, Pthreads, OpenMP and MPI [10+15+10+10 = 45Marks

You need to be as specific as possible and provide the answers in the spac overwriting and ambiguous answers would be considered as incorrect. It is importa and cleanly provide the answer so think before you write anything.

Please make sure to properly comment your code and write clearly, failure to do s 10 marks. Code containing overwritten text and corrections cannot be claimed for

You have been asked to write code for different parts of the problem in pr
please note that the complete program should work as a whole, that is only code for some sub problem is not acceptable.

I. You need to write a program where the parent process initializes an array of 10 integers (3points) It then creates a child process which then finds the maximum number amongst the array (2points) (3points) The child process reports back to parent process, which displays the results (3points)

```
/* write the code to initialize the array (3points) */

int arr[10] = {1,2,3,4,5,6,7,8,9,10};

pid= fork();


/* write the code for the rest of the question (1+3+3 points) */
if(pid==0)    \\child
{
for(i=1; i<10; i++)   \\using sorting  for array containing 10 elements
{
if(arr[0]<arr[i])
{
arr[0]= arr[i];
}
}
int maximum= arr[0];
else  \\parent
{
wait(&status);
\\returns the exit status of the child
Printf(" The maximum number is : %d\n", WEXITSTATUS(status));
}
```

II. You need to write a program where the master thread initializes an array of n (you can initi n to any number) integers. It then creates n/2 threads (using pthread API) and passes an in number to each thread (3points) Each thread then sums up two numbers from the array (3points) (index and index+1, where index is passed by the master thread) and writes the sum to a s variable in a synchronized manner (4points)

```
/* write the code to initialize the array */
int *arr = (int *)malloc((n) * sizeof(int));
   printf("The element of arrays are: ");
   for (int i = 0; i < n; i++)
   {
      arr[i] = rand() % 11;
       printf("%d ", arr[i]); }
```

/ write the code to create n/2 threads (using pthread API) and pass an index number to each thread **(6points)***

```
int N,n;
   printf("Enter the number of threads: ");
  // scanf("%d", &N);
   n= N/2;
   printf("Number = %d",n);
   pthread_t th[n];
   int i;
   for (i = 0; i < n; i++) {
      int* a = malloc(sizeof(int));
      *a = i * 2;
      if (pthread_create(&th[i], NULL, &routine, a) != 0) {
         perror("Failed to create thread"); }
```

/ write the code for each thread to sum given numbers from the array **(3points)** and write it to the shared variable in a synchronized manner **(6points)***

```
int Sum = 0;
   for (i = 0; i < n; i++) {
      int* r;
      if (pthread_join(th[i], (void**) &r) != 0) {
         perror("Failed to join thread");
      }
      Sum += *r;
      free(r);
   }
   printf("Sum: %d\n", Sum);
```

III. You need to redo the question in previous part using OpenMP instead of pthreads. ignore the array initialization part and more specific details are given below.

/ write the code to create n/2 threads (using openMP) and somehow pass an index number to each thread **(2+3points)***

```
#pragma omp parallel num_threads(n/2)
```

/ write the code for each thread to sum given numbers from the array* ***(3points)*** *and write it to the shared variable in a synchronized manner* ***(2points)***

```
for(i=0; i<=n/2; i++)
{
sum= arr[i]+arr[i+1];
i=i+1;
#pragma omp critical
{
total_sum=total_sum+sum;
}
```

IV. You need to write a program, using MPI, where each process (amongst any n proce initializes a number to be its rank*2, process ranked 4 initializes the number to be process then **broadcasts** its number*, and* when the code completes each process h *numbers* of all the processes. For communication, you are allowed to use **MPI_Bcast only**.

/ write the code where each process initializes its number to be its rank*2* ***(4points)***

```
MPI_Init(&argc, &argv);
 MPI_Comm_size(MPI_COMM_WORLD, &size);
 MPI_Comm_rank(MPI_COMM_WORLD, &rank);
 MPI_Get_processor_name(processorName, &nameLen);

for(i=0; i<rank_num.size(); i++)
{
rank_num[i]=rank*2;
}
```

/ write the code where each process broadcast its number to all others, that is, when the code executes each process has the localNumbers of all process.* ***(6points)***

```
for(i=0; i<rank_num.size(); i++)
{
MPI_Bcast(rank_num[i], 13, MPI_INT, i, MPI_COMM_WORLD);
printf( " Message from process %d\n", rank_num[i]);
}
```