Md. Ashikur Rahman
183110

1. (a)(ii) Heap memory

~~(iii) Root Global variable~~

(b) True

(c) The difference between thread local storage and static data is that thread local storage data are are unique to each other.

(d) Data sharing between threads belonging to the same process occurs easily in windows and Pthreads, since share data are simply declared globally. As Java has no such notion of global data. We can pass parameters to class that implement Runable, but Java threads cannot return results. Results returned from callable task are known as

Future object. A result can be retrieved from callable method defined in Future interface. ~~The Program shows~~

(e)
```
Public class sumtask exitude recursivetask
                        <integer> {

Static first int THRESHOLD=50;
Private int begin;
private Int end;
Private int [*] array;
Public sum take (int begin, int end; int []
                                     arr] {
 this. begin = begin;
 this. end = end;
 this. array = array
Protectad integer compute () {
if (end, begin < Threshold) }
int sum = 0
```

```
for ( int i = begin; i<=end;
i++) {
sum+ = arr[i]; }
else { int mid (begin+end)/2
sumTask cosfund — ne (b, m, a),
```

2. (a) All of the above

(b) False

(c) Three IPC message —

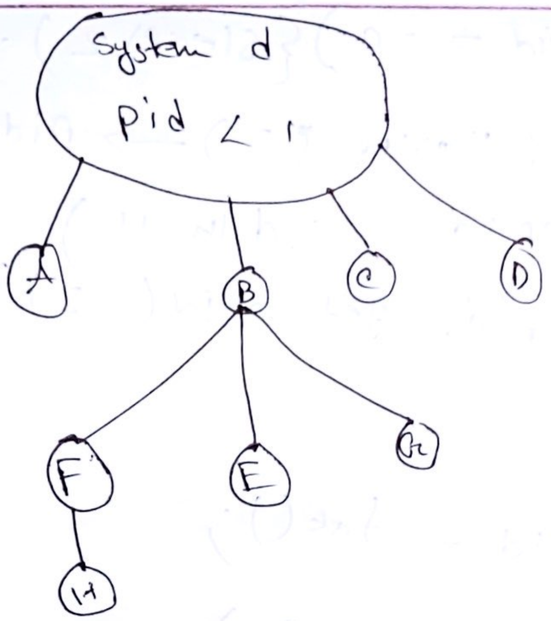
(i) For small message (up to 256 bytes), the port message queue is used as intermediate stroage and message are copied from one processs to other.

(ii) larger messages must be passed through a section object, which is a region of shared memory associated with the channel.

(iii) when the amount of data is too large to fit into a section object An API is available that allows server processes to read and write directly into the adress space of client.

(d) A Race condition is a situation in which two or more threads or processes are reading or writing some shared data, and the final result depends on the timing of how the threads are ~~sh~~. scheduled. Race condition can lead to unpredictable result and subtle program bugs. Race conditions in the producer - consumer example prevented by having the stroage of a new integer into the CubbyHole by the producer by synchronized with the retrieval of an integer from the cubbyHole by the consumer.

3. (e)
2



```
int main() {
  int pid, pid1, pid2 ;
    pid = fork ( );
 if (pid = =0) { sleep<3 );
 print ["child(F)"]
   pid = %d and
 ppid = %d  \n"
 Get  pid (),  Get  ppid ') ; }

 if else
  { pid1 = fork ( );
```

```
if ( pid = =0 ){ sleep(2 ) ;
Print ( "child [2) —> pid = %. d
End ppid = %d \n " ).
get pid. get ppid( 2)}

else
 { pid = fak();
if  (pid 2==0 )
 { Print ("child (3))
       get pid (e) }
```

3. (b) All of above

(a) False

(c) Every task has aceess to a bootstrap port which allows a task to register a port it has created with a system with bootstarup server. Once a port has been registered withe the bootstrap servers, other task can look up the port in this registry and obtain right for sendin message to the port.

3. (d) To find the port of a remote program, a client sends and RPC call message to a server portmap. If the remote program is register with the portmap it returns the relevant port number in a RPC reply message. The client program can then send RPC call message packets to that remote program port.

3. (e)

line x	line y	line z
child = 16;	child = - 16	parcent ~~present~~ = 0
child = 36;	child = - 36	parcent = 4
child = 64	child = 64	Parcent = -10
child = 100	child = - 100	Parcent = 8
child = 144	child = - 104	Parcent = - 4