

PART 1

1) What are threads and multithreading?

⇒ Thread provides the ability to execute multiple transaction in the same transaction environment

⇒ Thread used to decode difficulties in process sharing

⇒ Multithreading is ability to provide multiple execution threads that are supported by the OS at the same time.

2) Threads require fewer resources and generate less overhead but process need to special address and memory also take up space.
(memory allocation)

Threads share the memory that is allocated to the process.

For these reasons, thread is faster than process

3) What are the advantages of threads in OS?

⇒ it can perform multiple operations in one process

⇒ It require fewer resources they are easy to create and destroy

⇒ Threads share the memory

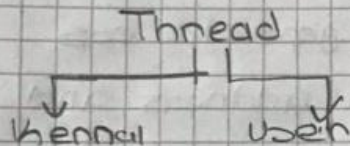
⇒ It is very easy to make memory shares

4) What is the difference between a process and a thread?

⇒ Threads are located in the process and share the same address space. They can carry out more than one job in one process.

⇒ Threads are not independent of each other like processes. They share resources and some data each other among themselves.

5) What are the types of threads? Explain the types



A user thread is one that executes user-space code.

But it can call into kernel space at any time. It's still considered a "User" thread, even though it's executing kernel code at elevated security levels.

A Kernel Thread is one that only runs kernel code and isn't associated with a user-space process.

USER

Three primary thread libraries;

1. POSIX

2. Windows

3. Java

KERNAL

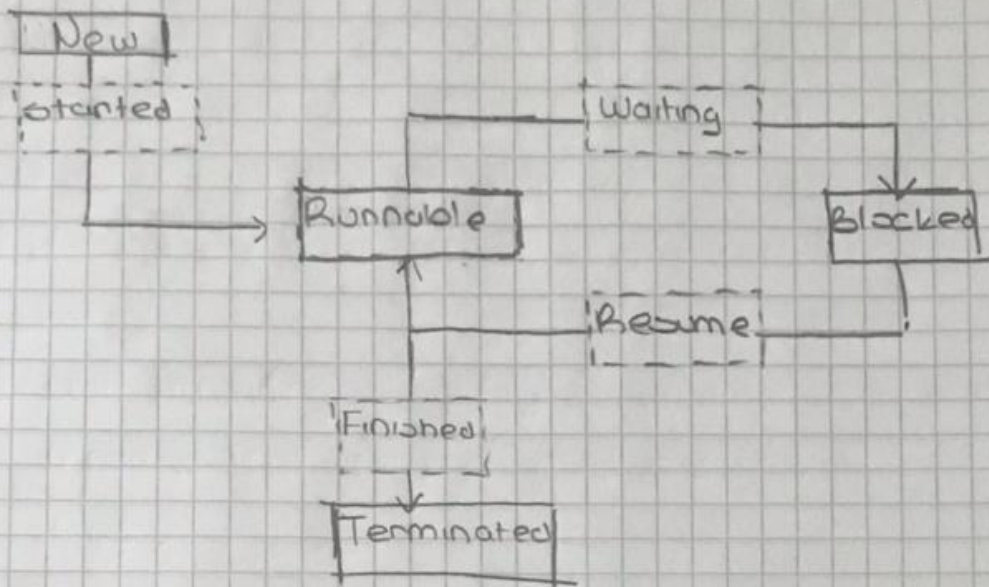
Windows

Solaris

Linux

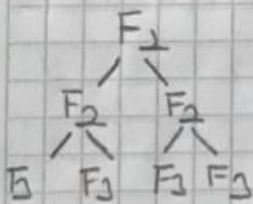
Mac OS X

6) What are the states of threads in OS?



7)

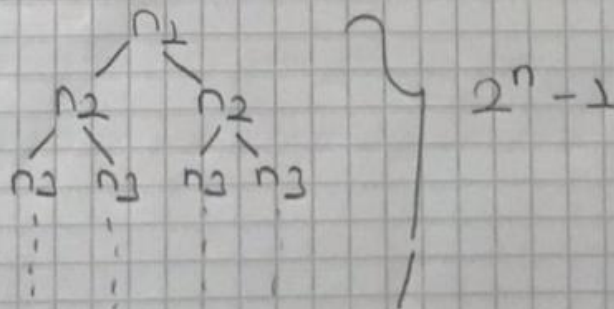
Fork() is a system call that creates a new process. After a new child process is created, both processes will execute the next instruction.



$$2^n - 1$$

$$2^3 - 1 = 7$$

8) $2^n - 1$



$$2^n - 1$$