

CS162, Spring 2002
Section 2 Quiz 1
Steve Martin

1. What is a Process?

A single stream of execution running in its own address space along with everything we need to execute the program.

This includes:

Code they're running

STATE (contained in address space)

Set of Registers

Address Space

Program Counter (controls stream of execution)

Stack Pointer (where we're at in our memory)

Resources

2. What's the difference between a process and a program?

Processes have state, which includes many things not found in code alone. A program is just typically some kind of code executable after compilation or through interpretation. A process is a running program, with all the state that entails.

3. Why is a thread sometimes called a 'lightweight process'?

Threads are cheaper to create than processes in that they primarily share a lot of state with other threads within the same process. As an example, a thread shares program code, process resources, etc.

4. Why use threads at all?

Using threads can give the illusion of doing things concurrently within the same process. Example: clicking on things in Netscape while the page is still loading.

Also, creating a thread entails much less overhead than creating a new process!

5. What is shared and what is separate among threads?

Shared:

Code

Address Space

Resources

Separate:

Program Counter

State

Stack

Registers (included in state)