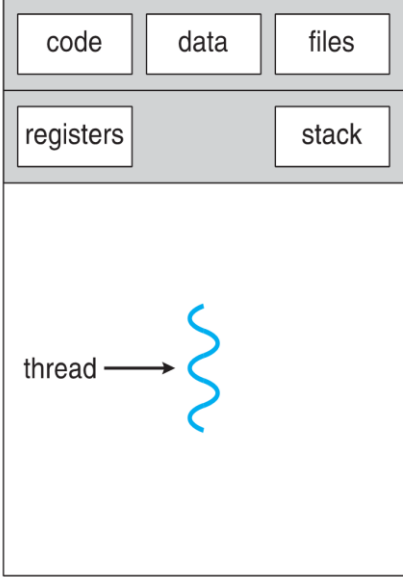
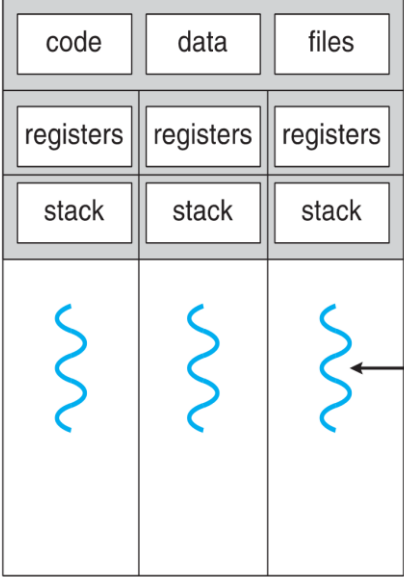
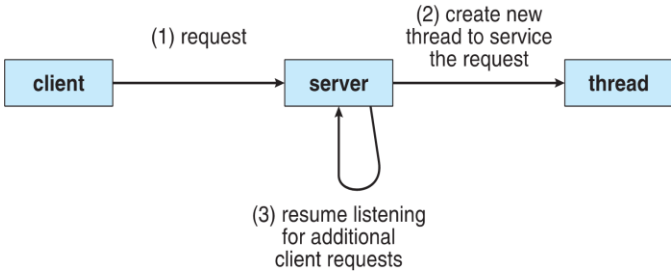


Threads	<ul style="list-style-type: none"> <li>• What is a thread? <ul style="list-style-type: none"> <li>○ Basic unit of CPU utilization</li> <li>○ Comprised of <ul style="list-style-type: none"> <li>▪ Thread ID</li> <li>▪ Program counter</li> <li>▪ Register set</li> <li>▪ Stack</li> </ul> </li> </ul> </li> <li>• Multiple threads can run in a process and share: <ul style="list-style-type: none"> <li>○ Code section</li> <li>○ Data section</li> <li>○ OS resources (e.g., open file)</li> </ul> </li> </ul> <div style="display: flex; justify-content: space-around; align-items: flex-end;"> <div style="text-align: center;">  <p>single-threaded process</p> </div> <div style="text-align: center;">  <p>multithreaded process</p> </div> </div> <ul style="list-style-type: none"> <li>• Traditional (Heavyweight) process has a single thread of control</li> <li>• Multithreaded process can do more than one task at a time</li> </ul>
Threads – Application Example	<ul style="list-style-type: none"> <li>• Web Browser with multiple threads: <ul style="list-style-type: none"> <li>○ Retrieve data from the network</li> <li>○ Display images</li> <li>○ Display text</li> </ul> </li> <li>• Web server <ul style="list-style-type: none"> <li>○ Accepts client requests for images, sound, etc.</li> </ul> </li> </ul>
<div style="text-align: center;">  <pre> sequenceDiagram     participant client     participant server     participant thread     client-&gt;&gt;server: (1) request     server-&gt;&gt;thread: (2) create new thread to service the request     thread-&gt;&gt;server: (3) resume listening for additional client requests           </pre> </div>	

Why Threads?	<ul style="list-style-type: none"> <li>• Threads act as part of a process instead of creating an entirely new process</li> <li>• Shared access to code, data, and files</li> </ul>
Thread Benefits	<ul style="list-style-type: none"> <li>• Responsiveness</li> <li>• Resource sharing</li> <li>• Economy               <ul style="list-style-type: none"> <li>◦ Solaris OS creating a thread is 30 times faster</li> <li>◦ Solaris OS context switching is 5x faster with threads</li> </ul> </li> <li>• Scalability               <ul style="list-style-type: none"> <li>◦ Can run threads on multiple processors</li> </ul> </li> </ul>
Multicore Programming	<ul style="list-style-type: none"> <li>• Parallelism – can perform more than one task simultaneously</li> <li>• Concurrency – allows all tasks to make progress by rapidly switching between processes on a single CPU</li> </ul> <div> <ul style="list-style-type: none"> <li>• Amdahl's Law (S is the percent performed Serially, N is the number of processing cores):</li> </ul> <math display="block">speedup \leq \frac{1}{(S + \frac{(1-S)}{N})}</math> <ul style="list-style-type: none"> <li>• Modern computers with increased hardware enhancements may render this law irrelevant</li> </ul> </div>
Programming Challenges	<ol style="list-style-type: none"> <li>1. Identifying tasks – which tasks can be run independently and thus in parallel?</li> <li>2. Balance – is it worth the cost to separate some tasks?</li> <li>3. Data splitting – data accessed by tasks must be divided to run on separate cores</li> <li>4. Data dependency – must ensure synchronization where required</li> <li>5. Testing and debugging – more difficult given the inconsistency of run-time execution</li> </ol>
Types of Parallelism	<ol style="list-style-type: none"> <li>1. Data parallelism – how to split up the data?               <ol style="list-style-type: none"> <li>a. Example: sum an array</li> </ol> </li> <li>2. Task parallelism – split the work (tasks)</li> </ol>
Thread Libraries	<ul style="list-style-type: none"> <li>• Three main thread libraries               <ul style="list-style-type: none"> <li>◦ POSIX Pthreads</li> <li>◦ Windows</li> <li>◦ Java                   <ul style="list-style-type: none"> <li>▪ Windows – typically uses Windows API</li> <li>▪ Unix/Linux – typically implemented using Pthreads</li> </ul> </li> </ul> </li> <li>• Two general strategies for creating multiple threads               <ul style="list-style-type: none"> <li>◦ Asynchronous</li> <li>◦ Synchronous</li> </ul> </li> </ul>
Pthreads	<ul style="list-style-type: none"> <li>• Pthreads – POSIX standard defining an API for thread creation and synchronization               <ul style="list-style-type: none"> <li>◦ Specification, not an implementation</li> <li>◦ IEEE 1003.1c</li> </ul> </li> </ul>