Processes

ECE 454 / 751: Distributed Computing

Instructor: Dr. Wojciech Golab

wgolab@uwaterloo.ca

Slides are derived from A. S. Tanenbaum and M. Van Steen,
Distributed Systems: Principles and Paradigms, 2nd Edition, Pearson-Prentice Hall, 2006.

Learning objectives

To review basic OS-related concepts:

- Linux shell commands for controlling processes
- Context switching
- Threads
- Virtualization

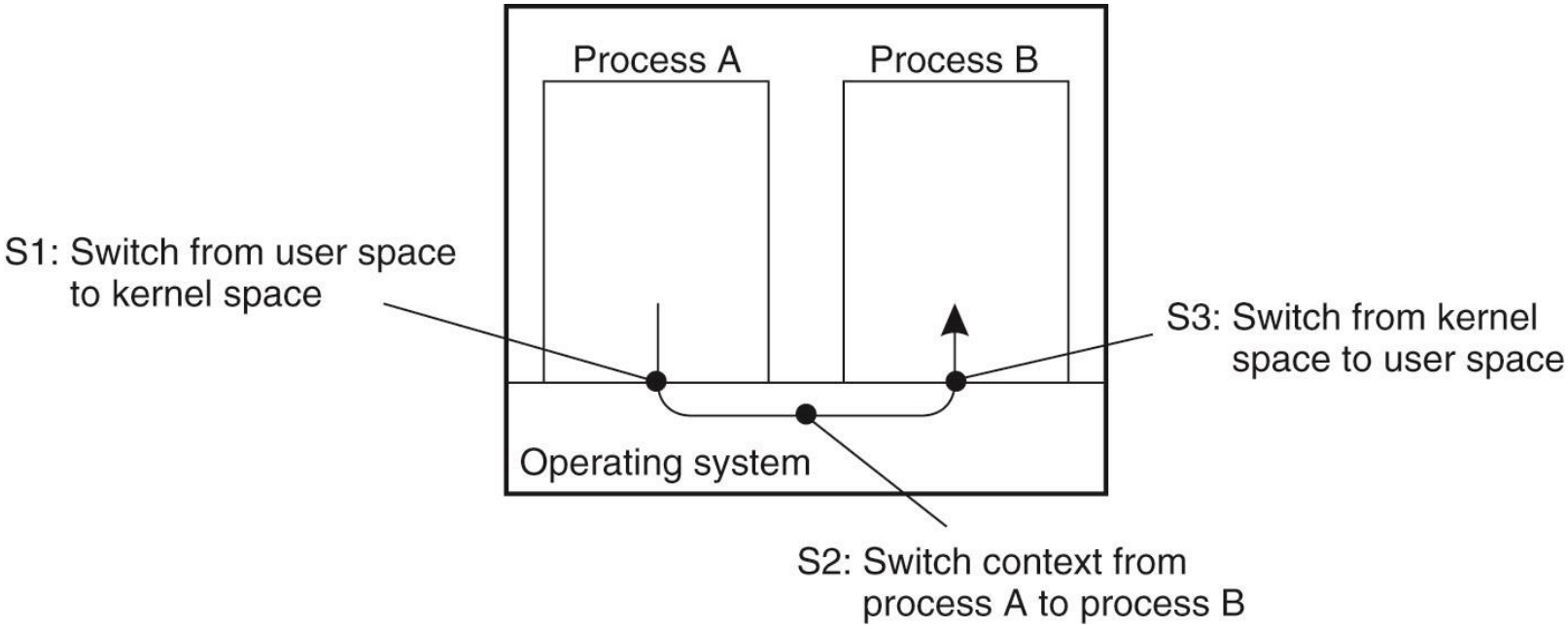
To develop a conceptual understanding of message passing protocols and server clusters.

Controlling processes in Linux

- ps – lists running processes
- top – displays processes with top resource usage
- kill / pkill / killall – terminates a process
- jobs – lists currently running jobs
- bg – backgrounds a job
- fg – foregrounds a job
- nice / renice – sets the priority of a process
- CTRL-C – stops a job running in a terminal
- CTRL-Z – suspends a job running in a terminal (use fg or bg to resume)

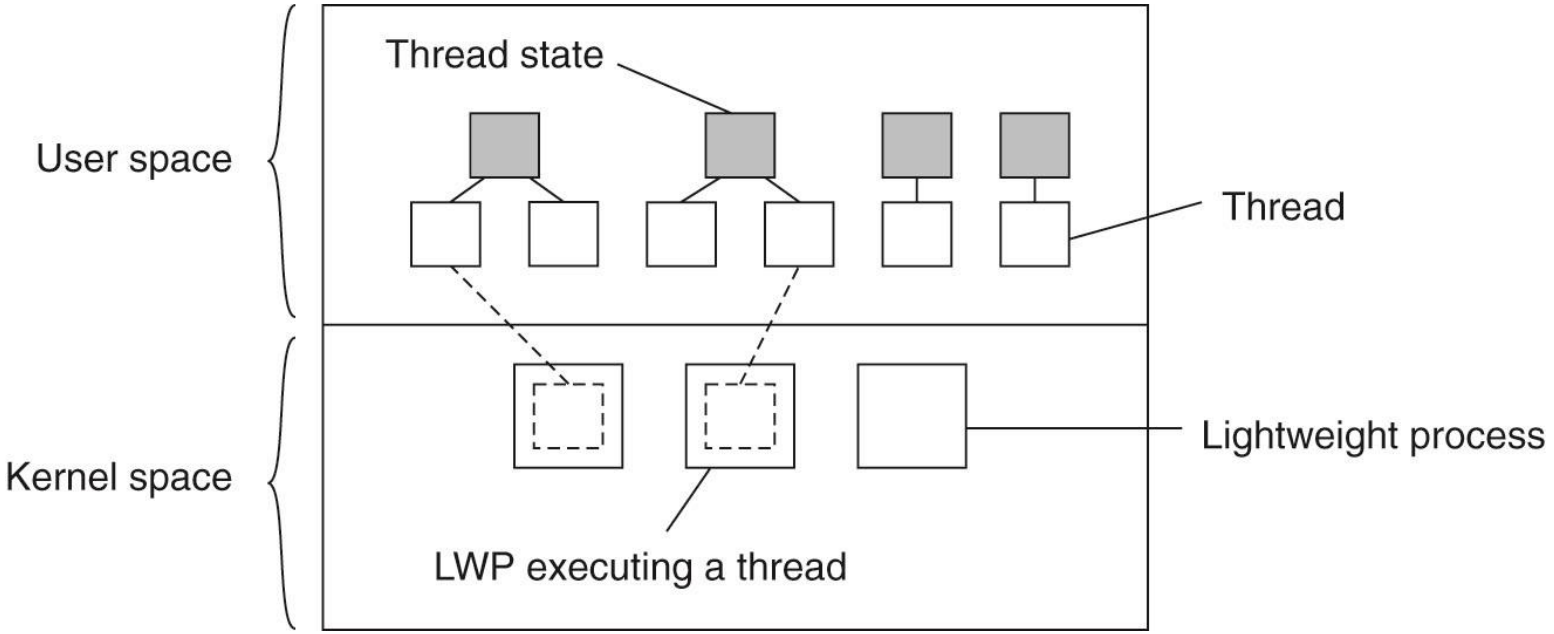
Context switching during IPC

Inter-process communication (IPC), such as using a named pipe in Unix/Linux, is costly in the sense that it requires a context switch from user space to kernel space and back.



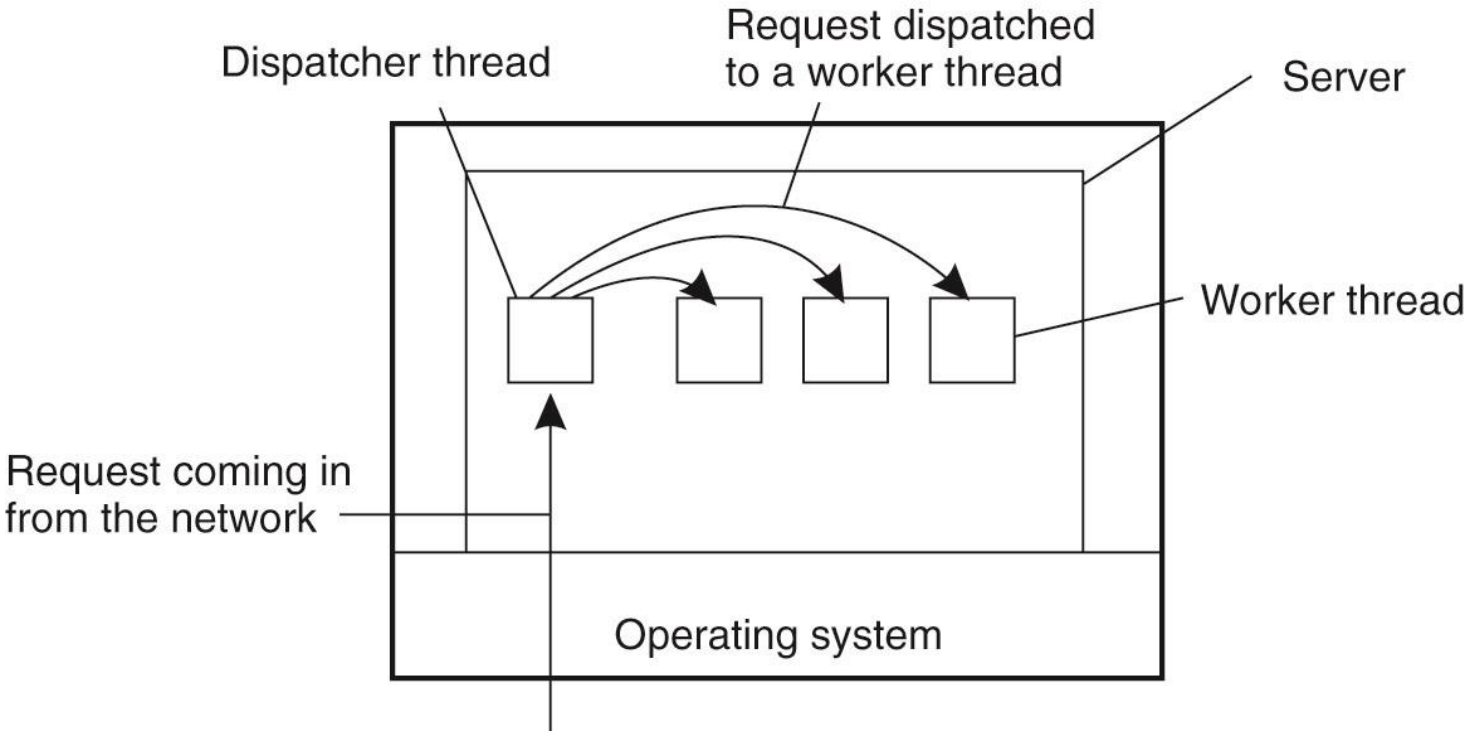
Threads

Threads executing in the same process can communicate directly through shared memory. An operating system kernel may support multi-threading through **lightweight processes (LWPs)**, which share an address space and file descriptors.



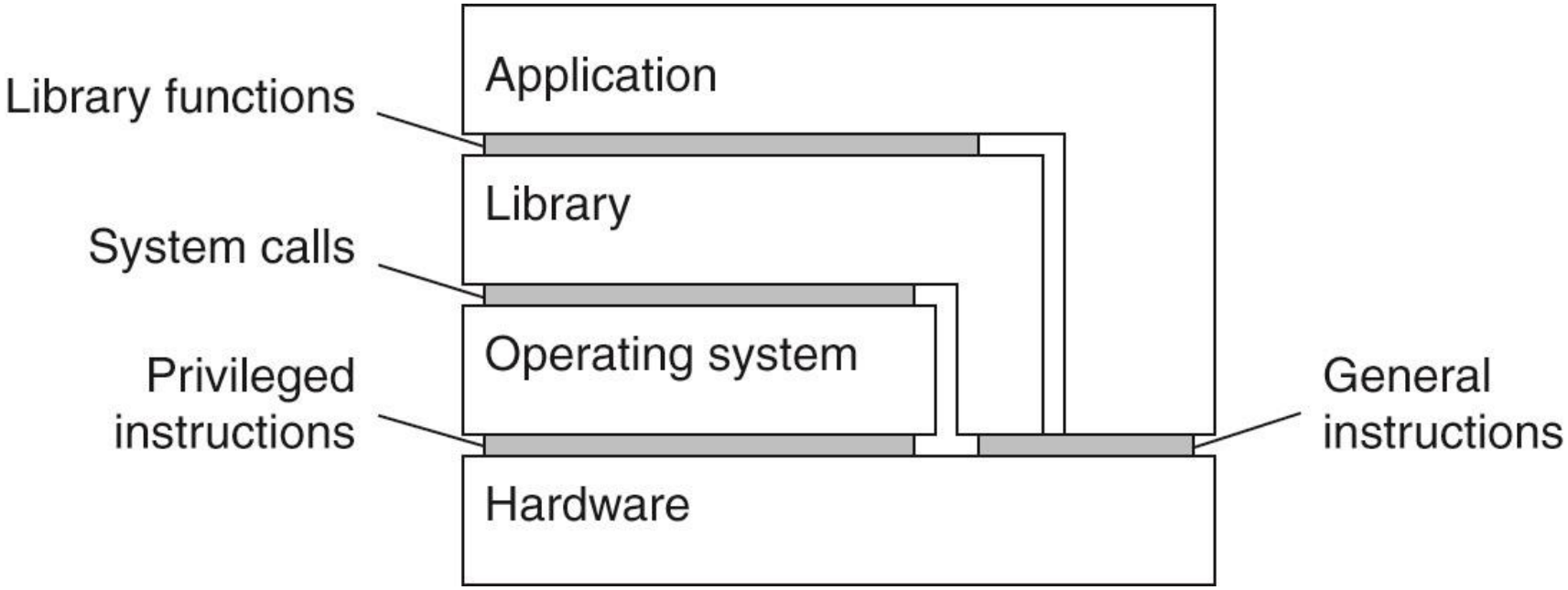
Multi-threaded servers

Many multi-threaded servers follow a **dispatcher/worker** design. The dispatcher thread receives requests from the network and feeds them to a pool of worker threads.



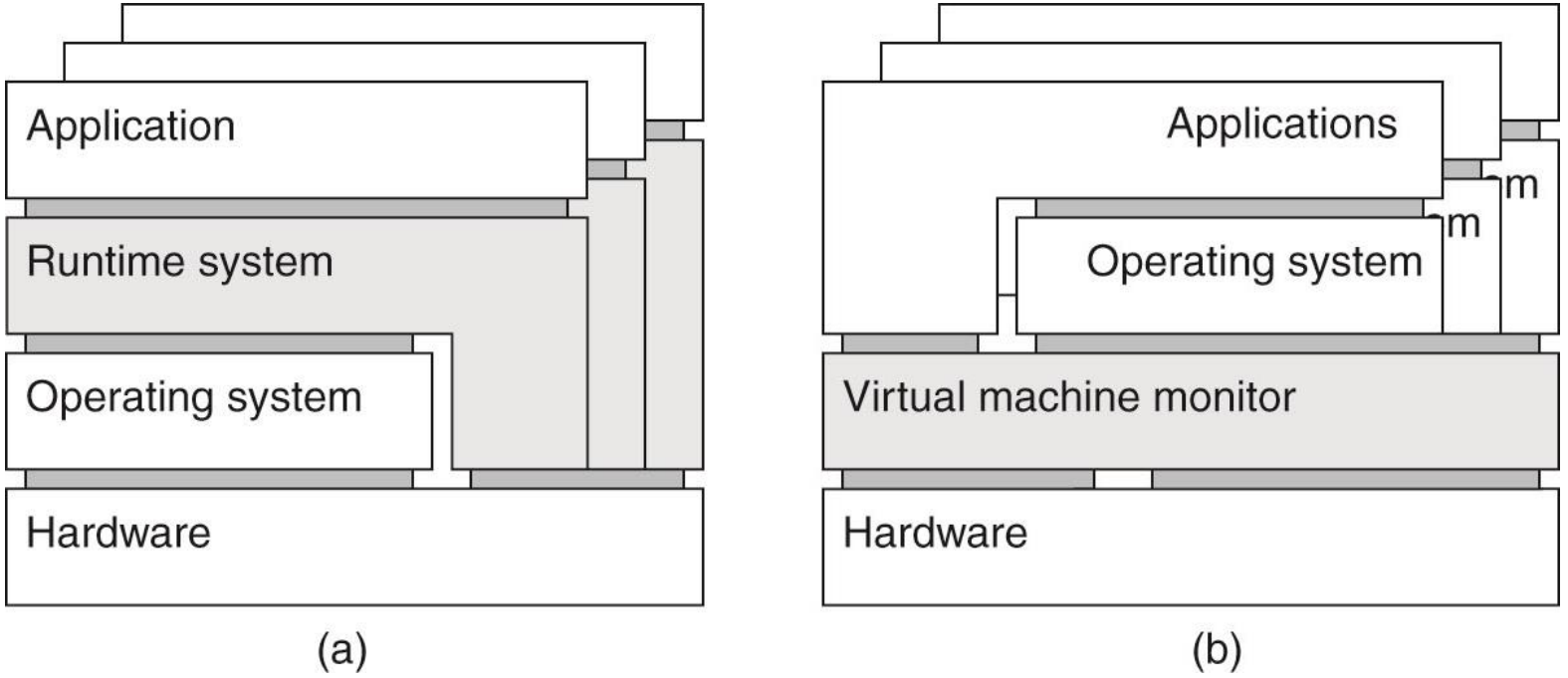
Hardware and software interfaces

Processes and threads interact with hardware either directly through processor instructions or indirectly through library functions and operating system calls.



Virtualization

Distributed system components often run in a virtualized environment such as a Java Runtime Environment (JRE, left image) or a virtual machine monitor (VMM, right image).



Virtualization

Virtualization tends to improve portability by isolating applications from the underlying hardware platform and, in some cases (e.g., JRE), the operating system as well.

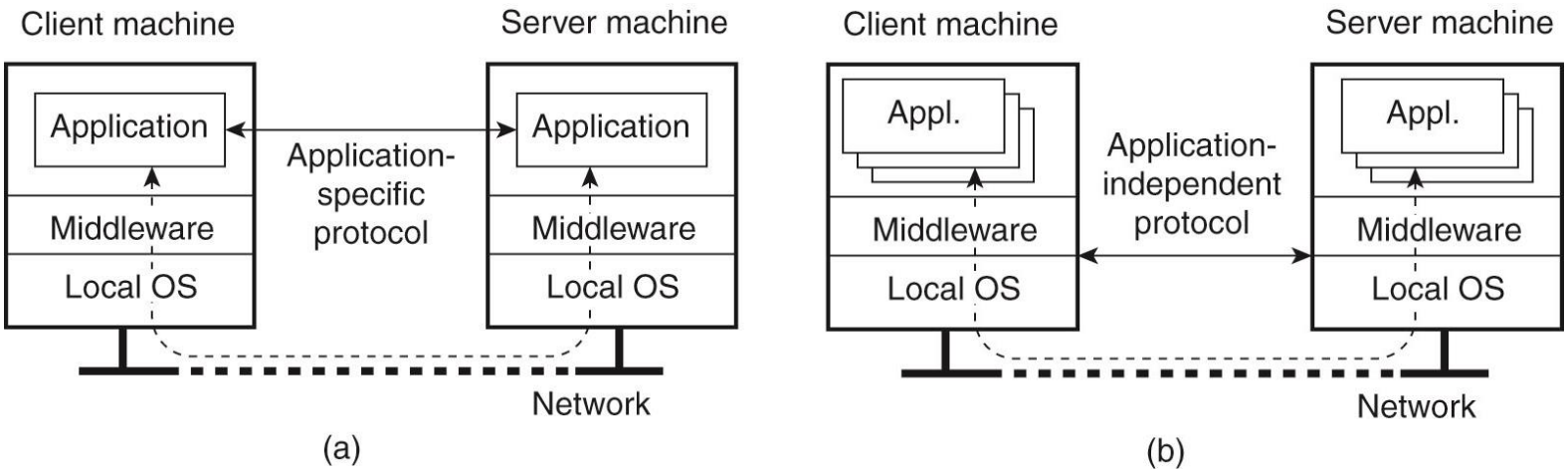
VMMs offer a number of additional benefits:

- server consolidation reduces capital and operating costs
- live migration of virtual machines (VMs) enables load balancing and facilitates proactive maintenance
- VM replication improves availability

Food for thought: what are the drawbacks of virtualization?

Interfaces in networked systems

Networked applications generally communicate by exchanging messages, for example using a middleware layer. A **message passing protocol** determines the format of the messages (e.g., binary versus human-readable text) and may be application-specific (a), or application-independent (b).



Server clusters

Servers in a cluster are often organized into three physical tiers. The first tier is a load balancer, which can be a specialized hardware device or a software component.