Introduction to Computer Science
Lesson 8

BSc in Computer Science
University of New York, Tirana

Assoc. Prof. Marenglen Biba

Operating Systems

- Lesson 6
  - The History of Operating Systems
  - Operating System Architecture
  - Coordinating the Machine's Activities
- Today
  - Handling Competition Among Processes
  - Security

Competing Processes

- An important task of an operating system is the **allocation** of the machine's resources to the **competing processes** in the system.
- Here we are using the term resource in a broad sense, including the machine's peripheral devices as well as features within the machine itself.
  - The file manager allocates access to files as well and allocates mass storages pace for the construction of new files;
  - The memory manager allocates memory space;
  - The scheduler allocates space in the process table; and the dispatcher allocates time slices.

Semaphores

- Let us consider a time-sharing operating system controlling the activities of a computer with a single printer.
- If a process needs to print its results, it must request that the operating system gives it access to the printer's device driver.
- At this point, the operating system must decide whether to grant this request, depending on whether the printer is already being used by another process.
  - If it is not, the operating system should grant the request and allow the process to continue;
  - Otherwise, the operating system should deny the request and perhaps classify the process as a waiting process until the printer becomes available.
- After all, if two processes were given simultaneous access to the computer's printer, the results would be worthless to both.

Semaphores

- A properly implemented flag, as just described, is called a **semaphore**, in reference to the railroad signals used to control access to sections of track.
- In fact, semaphores are used in software systems in much the same way as they are in railway systems.
- Corresponding to the section of track that can contain only one train at a time is a sequence of instructions that should be executed by only one process at a time. Such a sequence of instructions is called a critical region.
- *The requirement that only one process at a time be allowed to execute a **critical region** is known as **mutual exclusion.***

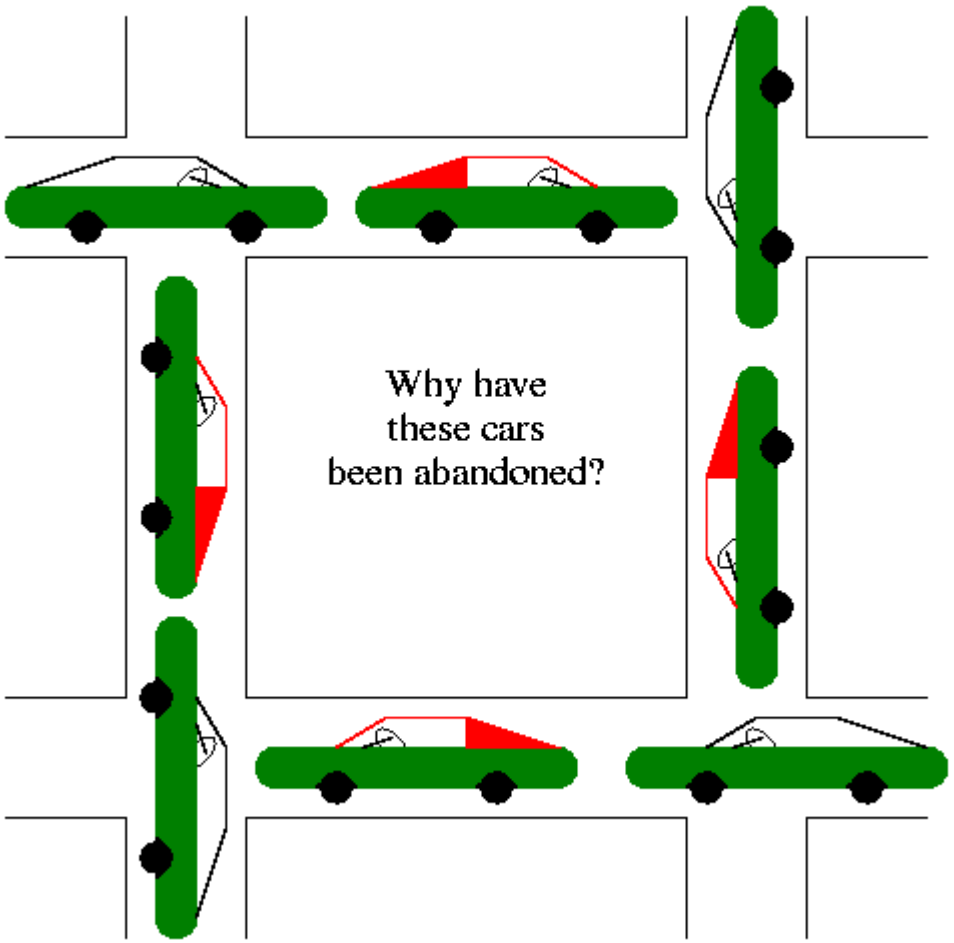
Deadlock

- Another problem that can arise during resource allocation is deadlock: the condition in which two or more processes are blocked from progressing because each is waiting for a resource that is allocated to another.

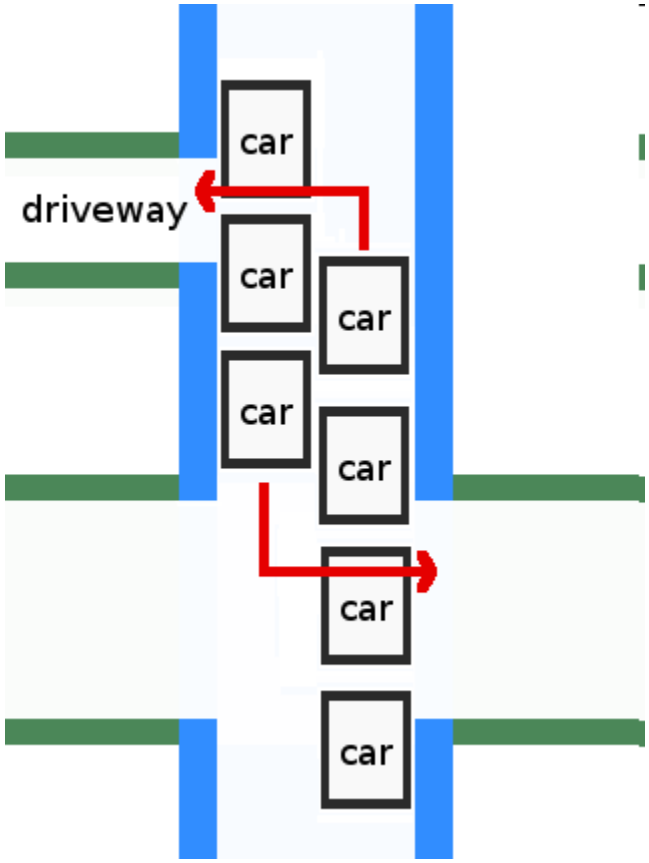
- For example, one process may have access to the computer's printer but be waiting for access to the computer's CD player, while another process has access to the CD player but is waiting for the printer.

- Such conditions, as in other settings can severely degrade a system's performance.

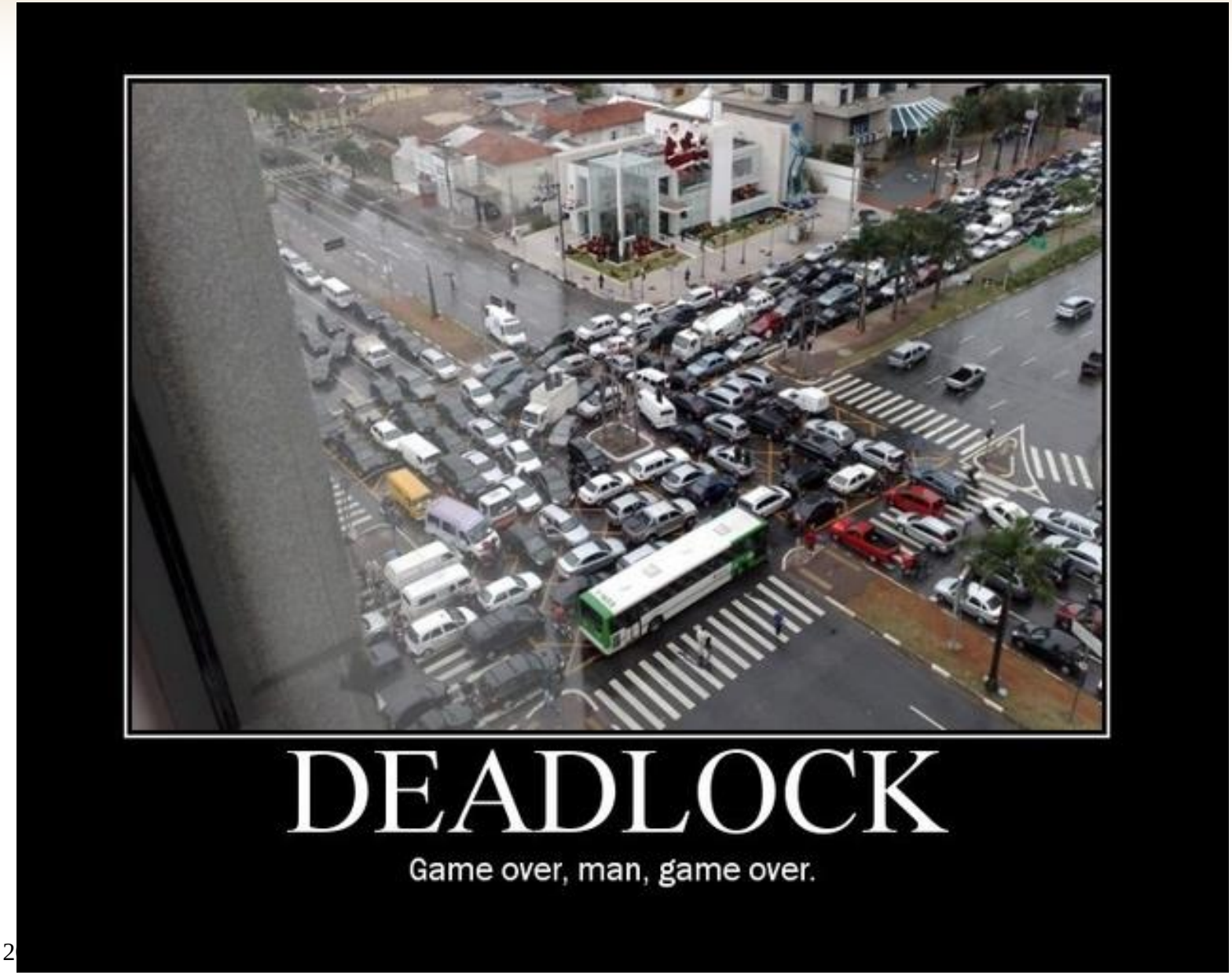
Funny deadlock



Another deadlock



Deadlock: Game over man!



Where do you go now?

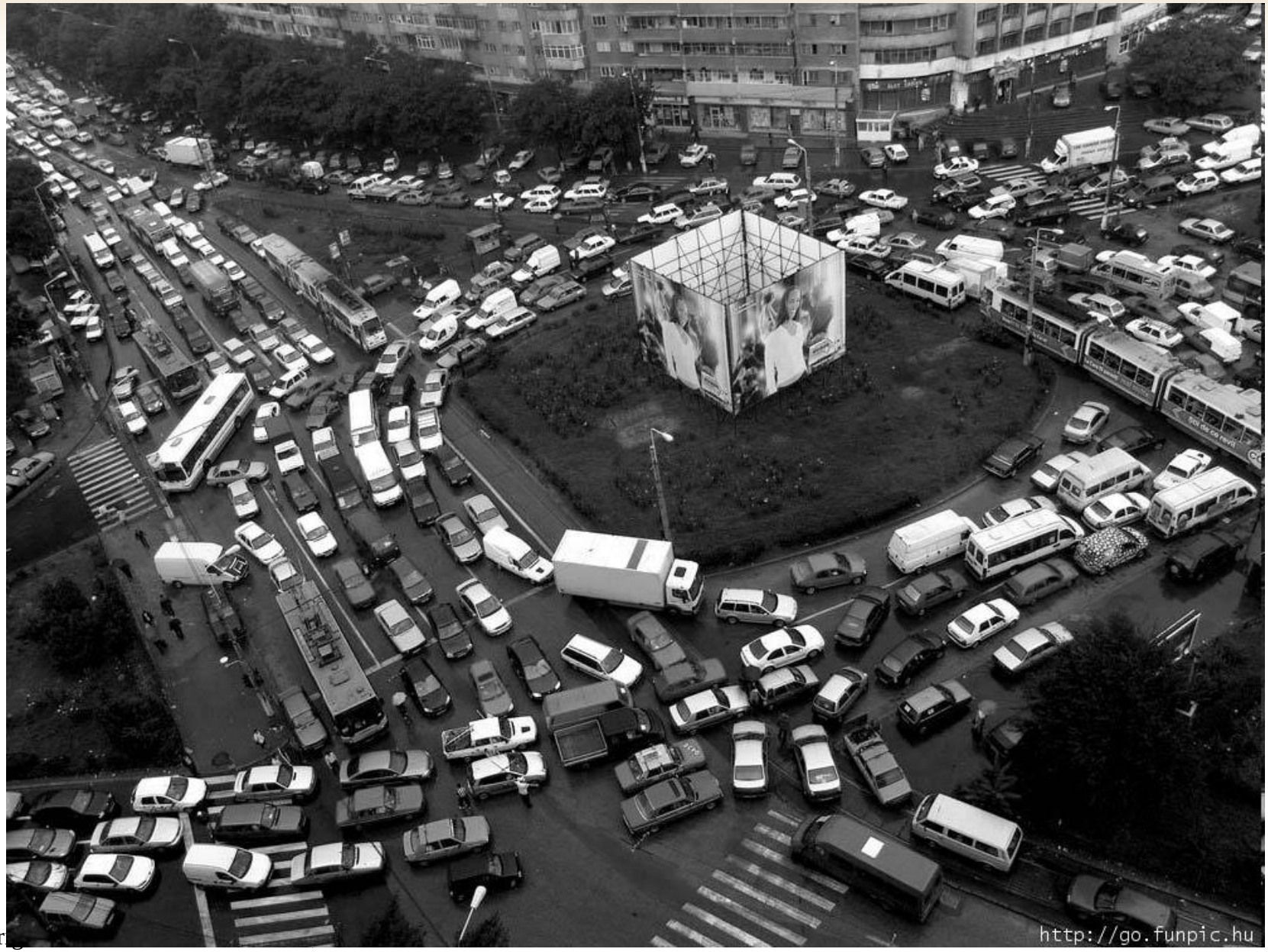
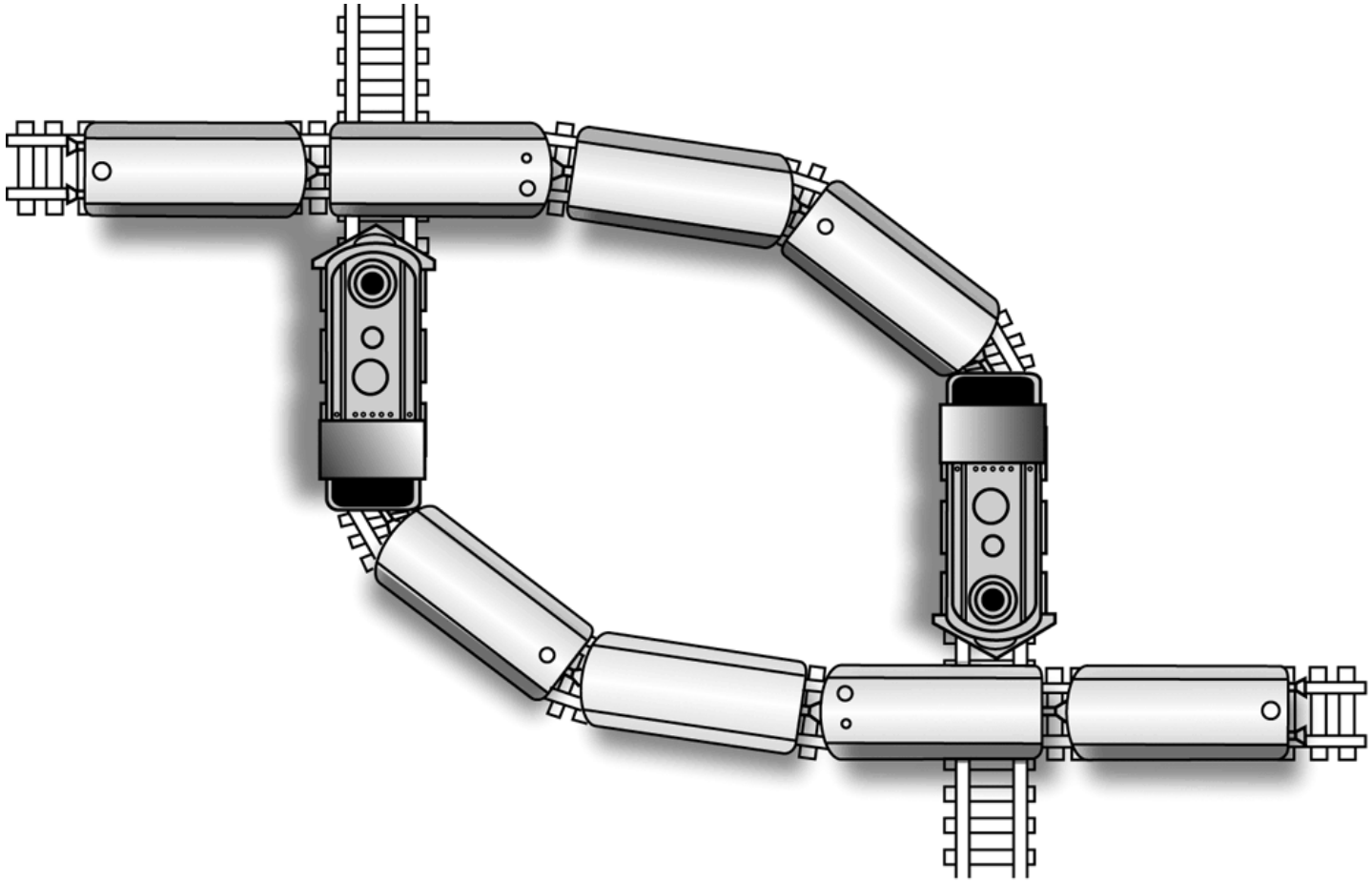
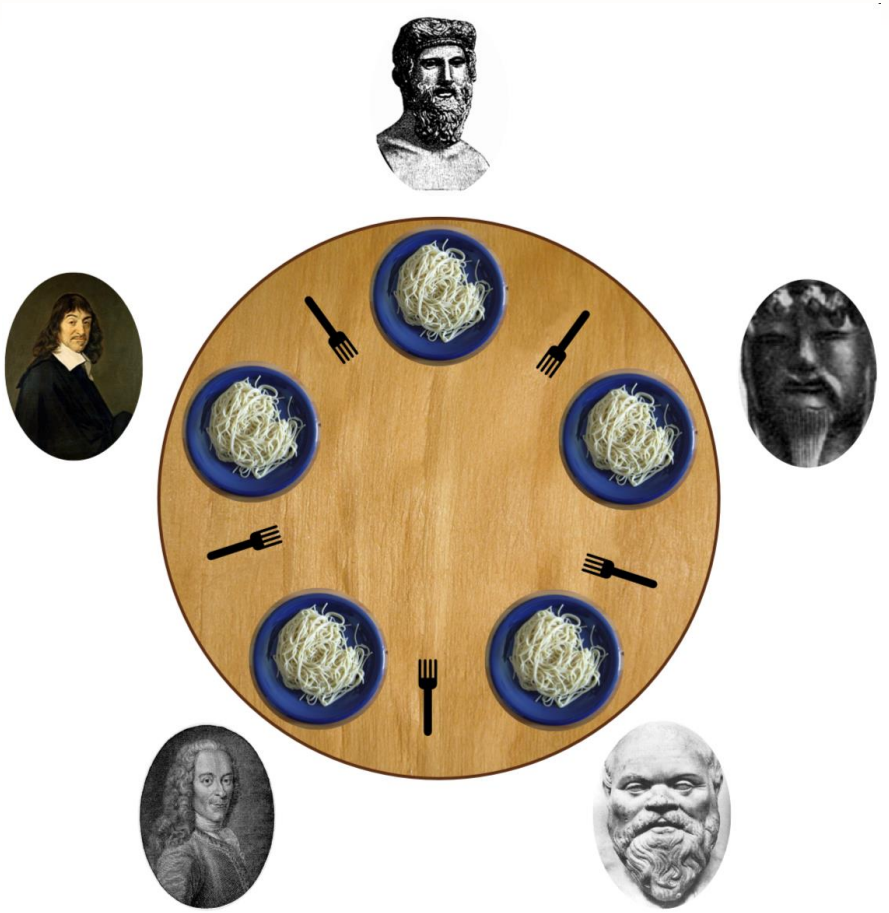


Figure 3.7 A deadlock resulting from competition for nonshareable railroad intersections



Dining-Philosophers Problem ☺

- Philosophers eat or think
- Eating needs 2 forks
- Pick one fork at a time
- How to prevent deadlock



Plato
Confucius
Socrates
Voltaire
Descartes

- Shared data
  - Bowl of rice (data set)
- Semaphore needed

Avoiding deadlocks

- Techniques that attack deadlock tend to be known as deadlock avoidance schemes.
- One, requires that each process requests all its resources at one time.
- Another, perhaps more imaginative technique attacks the first condition, not by removing the competition directly but by converting non shareable resources into shareable ones.

Operating Systems

- Lesson 6
  - The History of Operating Systems
  - Operating System Architecture
  - Coordinating the Machine's Activities
- Today
  - Handling Competition Among Processes
  - Security

Security

- Since the operating system oversees the activities in a computer, it is natural for it to play a vital role in maintaining security as well.
- In the broad sense, this responsibility manifests itself in multiple forms, one of which is reliability:
  - If a flaw in the file manager causes the loss of part of a file, then the file was not secure.
  - If a defect in the dispatcher leads to a system failure (often called a system crash) causing the loss of an hour's worth of typing, we would argue that our work was not secure.
- The development of reliable software is not a subject that is restricted to operating systems.
- It permeates the entire software development spectrum and constitutes the field of computer science known as software engineering.

Attacks from outside

- An important task performed by operating systems is to protect the computer's resources from access by unauthorized personnel.
- In the case of computers used by multiple people, this is usually approached by means of establishing "accounts" for the various authorized users
  - an account being essentially a record within the operating system containing such entries as the user's name, password, and privileges to be granted to that user.
- The operating system can then use this information during each login procedure (a sequence of transactions in which the user establishes initial contact with a computer's operating system) to control access to the system.

Super user and Administrators

- Accounts are established by a person known as the **super user** or the **administrator**.
- This person gains **highly privileged access** to the operating system by identifying himself or herself as the administrator (usually by name and password) during the login procedure.
- Once this contact is established, the administrator can alter settings within the operating system, modify critical software packages, adjust the privileges granted to other users, and perform a variety of other maintenance activities that are denied to normal users.

Auditing software

- The administrator is also able to monitor activity within the computer system in an effort to detect destructive behavior, whether malicious or accidental.
- To assist in this regard, numerous software utilities, called auditing software, have been developed that record and then analyze the activities taking place within the computer system.
- In particular, auditing software may expose a flood of attempts to login using incorrect passwords, indicating that an unauthorized user may be trying to gain access to the computer.
- Auditing software may also identify activities within a user's account that do not conform to that user's past behavior, which may indicate that an unauthorized user has gained access to that account.

Sniffing software

- Another culprit that auditing systems are designed to detect is the presence of sniffing software, which is software that, when left running on a computer, records activities and later reports them to a would-be intruder.
- An old, well-known example is a program that simulates the operating system's login procedure.
  - Such a program can be used to trick authorized users into thinking they are communicating with the operating system, whereas they are actually supplying their names and passwords to an impostor.

Role of Users: social engineering

- With all the technical complexities associated with computer security, it is surprising to many that one of the major obstacles to the security of computer systems is the carelessness of the users themselves.
  - They select passwords that are relatively easy to guess (such as names and dates),
  - they share their passwords with friends,
  - they fail to change their passwords on a timely basis,
  - they import unapproved software into the system that might subvert the system's security.
- For problems like these, most institutions with large computer installations adopt and enforce policies that catalog the requirements and responsibilities of the users.

Attacks from within

- Once an intruder (or perhaps an authorized user with malicious intent) gains access to a computer system, the next step is usually to explore, looking for information of interest or for places to insert destructive software.
- This is a straightforward process if the prowler has gained access to the administrator's account, which is why the administrator's password is closely guarded.
- If, however, access is through a general user's account, it becomes necessary to trick the operating system into allowing the intruder to reach beyond the privileges granted to that user.
- For example, the intruder may try to:
  - trick the memory manager into allowing a process to access main memory cells outside its allotted area,
  - or may try to trick the file manager into retrieving files whose access should be denied.

Defending from “attacks from within”

- Today's CPUs are enhanced with features that are designed to foil such attempts.
- As an example, consider the need to restrict a process to the area of main memory assigned to it by the memory manager.
- Without such restrictions, a process could erase the operating system from main memory and take control of the computer itself.
- To counter such attempts, CPUs designed for multitasking systems typically contain special-purpose registers in which the operating system can store the upper and lower limits of a process's allotted memory area.
- Then, while performing the process, the CPU compares each memory reference to these registers to ensure that the reference is within the designated limits.
  - If the reference is found to be outside the process's designated area, the CPU automatically transfers control back to the operating system (by performing an interrupt sequence) so that the operating system can take appropriate action.

Defending from “attacks from within”

- Without further security features, a process could still gain access to memory cells outside of its designated area merely by changing the special-purpose registers that contain its memory limits.

- That is, a process that wanted access to additional memory could merely increase the value in the register containing the upper memory limit and then proceed to use the additional memory space without approval from the operating system.

Privileged and non levels

- To protect against attacks, CPUs for multitasking systems are designed to operate in one of two privilege levels; we will call one "privileged mode" the other we will call "nonprivileged mode"
  - When in privileged mode, the CPU is able to execute all the instructions in its machine language.
  - However, when in nonprivileged mode, the list of acceptable instructions is limited.
- The instructions that are available only in privileged mode are called privileged instructions.
  - Typical examples of privileged instructions include instructions that change the contents of memory limit registers and instructions that change the current privilege mode of the CPU.

Privileged mode

- When first turned on, the CPU is in privileged mode.
- Thus, when the operating system starts at the end of the boot process, all instructions are executable.
- However, each time the operating system allows a process to start a time slice, it switches the CPU to non privileged mode by executing a "change privilege mode" instruction.
- In turn, the operating system will be notified if the process attempts to execute a privileged instruction, and thus the operating system will be in position to maintain the integrity of the computer system.

Privileged instructions

- Privileged instructions and the control of privilege levels is the major tool available to operating systems for maintaining security.
- However, the use of these tools is a complex component of an operating system's design, and errors continue to be found in current systems.
- A single flaw in privilege level control can open the door to disaster from malicious programmers or from inadvertent programming errors.
  - If a process is allowed to alter the timer that controls the system's time-sharing, that process can extend its time slice and dominate the machine.
  - If a process is allowed to access peripheral devices directly, then it can read files without supervision by the system's file manager.
  - If a process is allowed to access memory cells outside its allotted area, it can read and even alter data being used by other processes.

- Thus, maintaining security continues to be an important task of an administrator as well as a goal in operating system design.

Threats and viruses

Program threats

- Processes, along with the kernel, are the only means of accomplishing work on a computer.
- Therefore, writing a program that creates a breach of security, or causing a normal process to change its behavior and create a breach, is a common goal of crackers.
- In fact, even most nonprogram security events have as their goal causing a program threat.
- For example, while it is useful to log in to a system without authorization, it is quite a lot more useful to leave behind a back-door daemon that provides information or allows easy access even if the original exploit is blocked.

Trap door

- The designer of a program or system might leave a hole in the software that only she is capable of using.
- This type of security breach is called trap door.
- For instance, the code might check for a specific user ID or password, and it might circumvent normal security procedures.
- Programmers have been arrested for embezzling from banks by including rounding errors in their code and having the occasional half-cent credited to their accounts.
- This account crediting can add up to a large amount of money, considering the number of transactions that a large bank executes.

Trojan horse

- A code segment that misuses its environment is called a Trojan horse.
- A Trojan horse, or Trojan, is a non-self-replicating type of malware which appears to perform a desirable function but instead does something else, often including a backdoor allowing unauthorized access to the target's computer.
- These backdoors tend to be invisible to average users.
- Trojans do not attempt to inject themselves into other files like a computer virus.
- Trojan horses may steal information, or harm their host computer systems

Trojan horse operations

- A Trojan may give a hacker remote access to a targeted computer system. Operations that could be performed by a hacker on a targeted computer system may include:
  - Use of the machine as part of a botnet (e.g. to perform automated spamming or to distribute Denial-of-service attacks)
  - Crashing the computer
  - Blue screen of death
  - Electronic money theft
  - Data theft (e.g. retrieving passwords or credit card information)
  - Installation of software, including third-party malware
  - Downloading or uploading of files on the user's computer
  - Modification or deletion of files
  - Keystroke logging
  - Watching the user's screen
  - Viewing the user's webcam
  - Controlling the computer system remotely

Trojan horse: login emulator

- A variation of the Trojan horse is a program that emulates a login program.
- An unsuspecting user starts to log in at a terminal and notices that he has apparently mistyped his password.
- He tries again and is successful.
- What has happened is that his authentication key and password have been stolen by the login emulator, which was left running on the terminal by the thief.

Trojan horse: spyware

- Another variation on the Trojan horse is spyware.
- Spyware sometimes accompanies a program that the user has chosen to install.
- Most frequently, it comes along with freeware or shareware programs, but sometimes it is included with commercial software.
- The goal of spyware is to download ads to display on the user's system, create pop-up browser windows when certain sites are visited, or capture information from the user's system and return it to a central site.

Beast (Trojan horse)

- Beast is a Windows-based backdoor trojan horse, more commonly known in the underground hacking community as a Remote Administration Tool or RAT.
- It is capable of infecting versions of Windows from 95 to XP.
- Written in Delphi and released first by its author Tataye in 2002,

Beast (Trojan horse)

Once connected to the victim, Beast offered the following features:

- File Manager – along with browsing victim's directories it could upload, download, delete, or execute any file
- Remote Registry Editor
- Screenshot and Webcam capture utility
- Services, Applications, and Processes Managers, providing the ability of terminating or executing any of these
- Clipboard tool that could get currently stored strings
- Passwords tool capable of recovering any stored passwords in the victim's computer
- Power Options (e.g. shutdown, reboot, logoff, crash, etc.)
- Some tools mainly for creating nuisance (e.g. mouse locking, taskbar hiding, CD-ROM operator and locker, URL opener, wallpaper changer, etc.)
- Chat client providing communication between the attacker and the victim
- Other tools such as a Remote IP scanner, live keylogger, offline logs downloader, etc.
- Server Controls (e.g. server deleter, updater, terminator, info provider, etc.)

Viruses

- Another form of program threat is a **virus**.
- Viruses are self-replicating and are designed to "infect" other programs.
- They can modify or destroy files and cause system crashes and program malfunctions.
- A virus is a fragment of code embedded in a legitimate program.
- As with most penetration attacks, viruses are very specific to architectures, operating systems, and applications.
- Viruses are a particular problem for users of PCs.
- UNIX and other multiuser operating systems generally are not susceptible to viruses because the executable programs are protected from writing by the operating system.
- Even if a virus does infect such a program, its powers usually are limited because other aspects of the system are protected.

Viruses

- Note that many viruses belong to more than one category.
- **File**. A standard file virus infects a system by appending itself to a file. It changes the start of the program so that execution jumps to its code.
- **Boot**. A boot virus infects the boot sector of the system, executing every time the system is booted and before the operating system is loaded. It watches for other bootable media (that is, floppy disks) and infects them.
- **Macro**. Most viruses are written in a low-level language, such as assembly or C. Macro viruses are written in a high-level language, such as Visual Basic. These viruses are triggered when a program capable of executing the macro is run.

Antivirus

- Antivirus or anti-virus software is software used to prevent, detect and remove malware (of all descriptions), such as: computer viruses, hijackers, keyloggers, backdoors, trojan horses, worms, dialers, fraudtools, and spyware.

- Computer security, including protection from social engineering techniques, is commonly offered in products and services of antivirus software companies.

Encryption

- Encryption is fundamental to computer security.
- Encryption transforms data into something an attacker cannot understand.
- In other words, encryption provides a means to implement data confidentiality.
- In addition, encryption allows us to check whether data have been modified.
  - It thus also provides support for integrity checks.

Encryption

- Because it solves a wide variety of communication security problems, encryption is used frequently in many aspects of modern computing.
- Encryption is a means for constraining the possible receivers of a message.
- An encryption algorithm enables the sender of a message to ensure that only a computer possessing a certain key can read the message.
- Encryption of messages is an ancient practice, of course, and there have been many encryption algorithms, dating back to before Caesar.

Cryptography as a Security Tool

- There are many defenses against computer attacks.
- The broadest tool available to system designers and users is cryptography.
- In an isolated computer, the operating system can reliably determine the sender and recipient of all interprocess communication, since it controls all communication channels in the computer.

Cryptographic keys

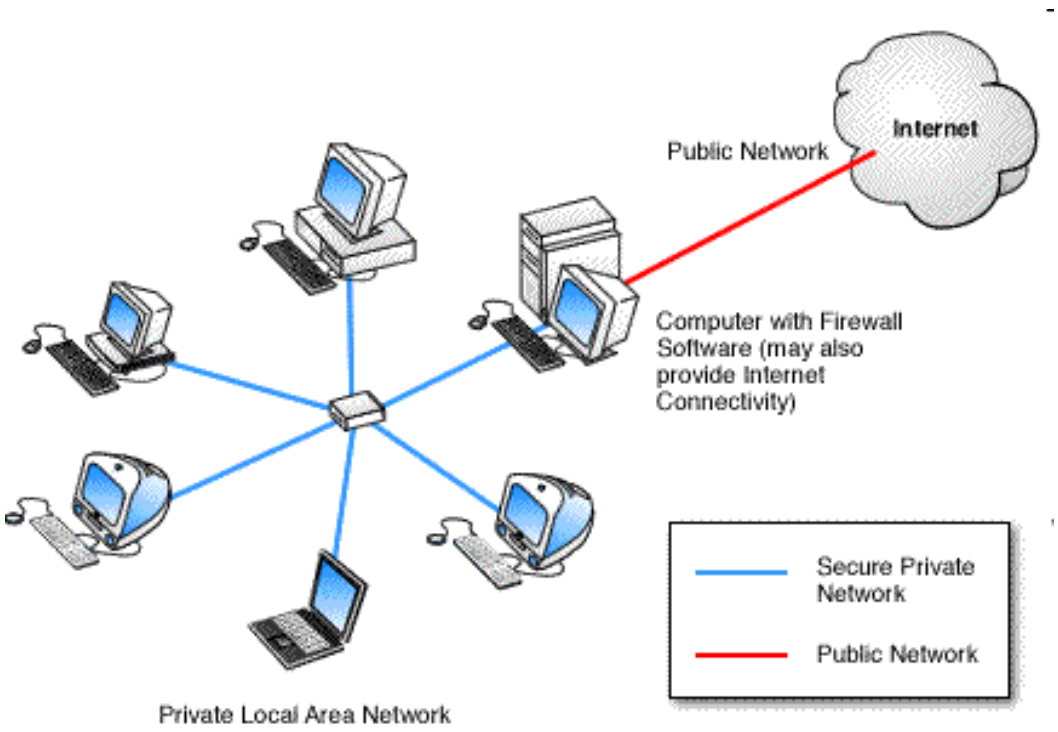
- Modern cryptography is based on secrets called **keys** that are selectively distributed to computers in a network and used to process messages.

- Cryptography enables a recipient of a message to verify that the message was created by some computer possessing a certain key- the key is the *source* of the message

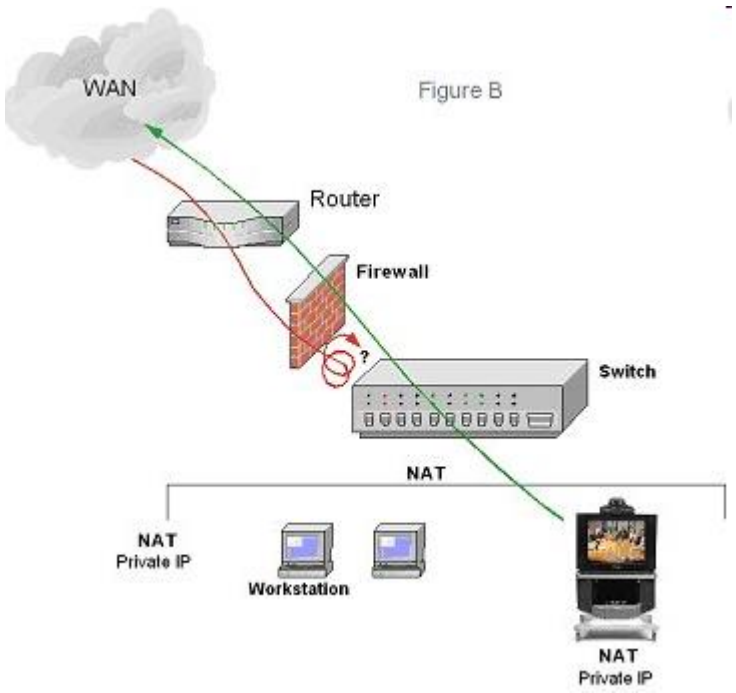
Firewall

- A firewall is a computer, appliance, or router that sits between the trusted and the untrusted.
- A network firewall limits network access between the two security domains and monitors and logs all connections.
- It can also limit connections based on source or destination address, source or destination port, or direction of the connection.

Software Firewall



Hardware Firewall



Personal firewall

- A personal firewall is a software layer either included with the operating system or added as an application.
- Rather than limiting communication between security domains, it limits communication to (and possibly from) a given host.

End of lesson 8

- Readings
  - Chapter 3

Appendix for home exercises: Linux commands

- cd → used for changing the current directory (in this example we are in the directory /Documents).

- mkdir → creates a new directory (ex: mkdir exampleDir)

ls

- ls → when used without any parameters, is used for listing the content of the current directory.
- ls –l → prints a detailed list of the content of the current directory
- ls –a → displays hidden files

rmdir

- rmdir → removes an existing empty directory using the directory name.
- rmdir –r →recursively removes a directory with all its content .
- rm –ir → removes a directory with all its contents but it prompts the user for each file that is being deleted.

cat, wc

- cat file(s) → send content of files in the standard output

- wc filename → counts lines, words and characters in the file. And displays them in the order we mentioned plus the name of the file.

cp

- cp file directory → copy file into a directory
- cp file directory/file1 → copy file at a directory but with a different name
- cp file1 file2 file... directory → copy multiple files in a directory

ps

- ps –ag → get information for all running processes
- pstree → display process information as a tree
- ps –u username → display the processes belonging to the specified user

top

- top → display currently running processes and additional information like memory and CPU usage with real time updates

grep

- grep searchString filename(s) → checks if a specific searchstring in the specified file(s). If the search string is found, the command displays the line in which the searchstring was found along with the file name.

- grep –i searchstring filename→ searches for a specific string in the file content by ignoring cases

diff

- diff –q filename1 filename2 → check the content of two provided files differ or not from each other

- diff filename1 filename2 → compares the content of the two provided files and then displays all lines of the files that do not match with each other

df

- df → displays information about the total disk space, the disk space currently in use, and the free space on all the mounted drives.
- df directory → displays the same info as above but it is limited only the drive on at which the directory is located.

free

- free → shows information about RAM and swap space usage, showing the total and the used amount in both categories
- -m → output is displayed in megabytes
- -k → output is displayed in kilobytes
- -b → output is displayed in bytes

Killall and kill

- Killall processname→ causes all processes under the processname to be killed and forced to finish execution

- Kill 0 → forcing all processes to stop except the shell
- Kill pID → kill processes by process ID
- Kill -9→ Sends a KILL signal with which the process really is annihilated by the operating system and make this process to finish execution.

history

- history → lists the last commands executed
- clear → clears the screen

sudo su (ubuntu)

- sudo su → change to root directory. It will prompt the user to enter the root password.

halt and reboot

- halt → to avoid loss of data we should use this command to shut down the program

- reboot → this command can be run only as root. It is the same as halt command but it will immediately reboot the system