Securing the operating systems

Bucharest, Romania

November 2019
Presented by

Alin Puncioiu
Ciprian-Ovidiu David

Part I

- Operating Systems
- Process Management
- Memory Management

Operating Systems

- In Kernel mode, the executing code has complete and unrestricted access to the underlying hardware. It can execute any CPU instruction and reference any memory address. Kernel mode is generally reserved for the lowest-level, most trusted functions of the operating system. Crashes in kernel mode are catastrophic; they will halt the entire PC.

- In User mode, the executing code has no ability to *directly* access hardware or reference memory. Code running in user mode must delegate to system APIs to access hardware or memory. Due to the protection afforded by this sort of isolation, crashes in user mode are always recoverable. Most of the code running on your computer will execute in user mode.

https://blog.codinghorror.com/understanding-user-and-kernel-mode/
https://en.wikipedia.org/wiki/Protection_ring

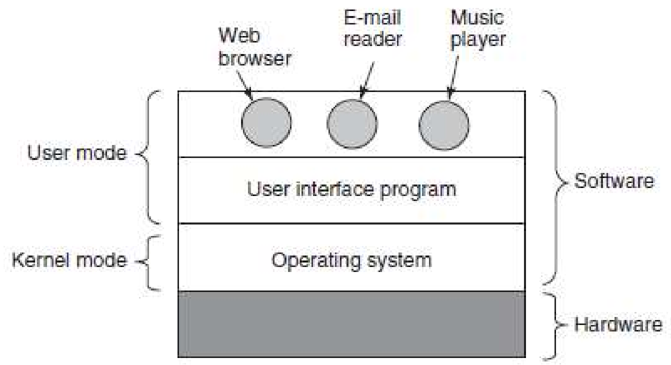


Figure 1-1. Where the operating system fits in.

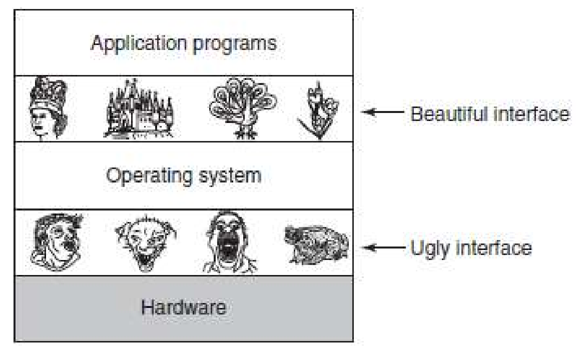
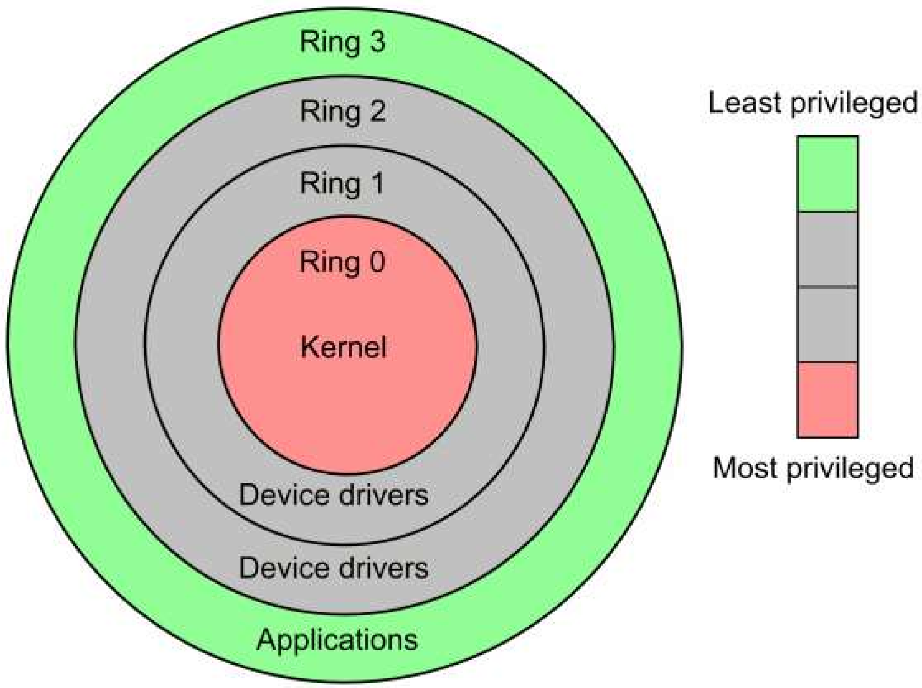


Figure 1-2. Operating systems turn ugly hardware into beautiful abstractions.

x86 CPU hardware actually provides four protection rings: 0, 1, 2, and 3. Only rings 0 (Kernel) and 3 (User) are typically used.



Process Management

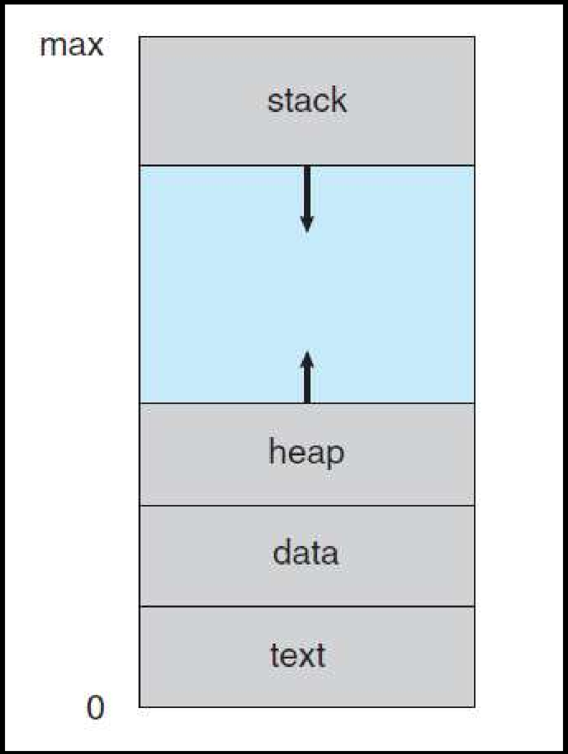
Processes

- A process is a program in execution

- A process generally also includes the process stack,
which contains temporary data (such as function parameters, return addresses, and local variables), and a data section, which contains global variables

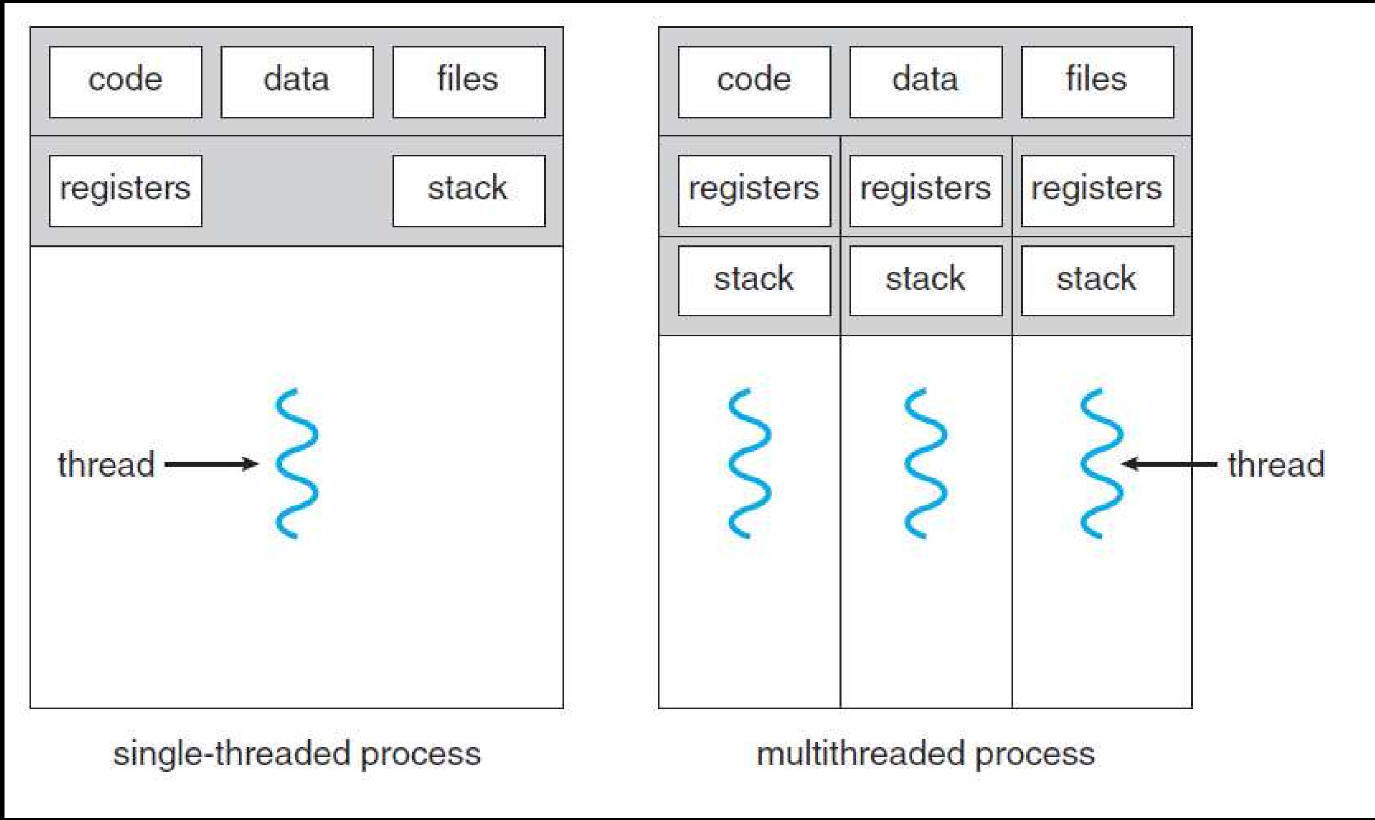
- A program is a passive entity, such as a file containing a list of instructions stored on disk(often called an executable file)

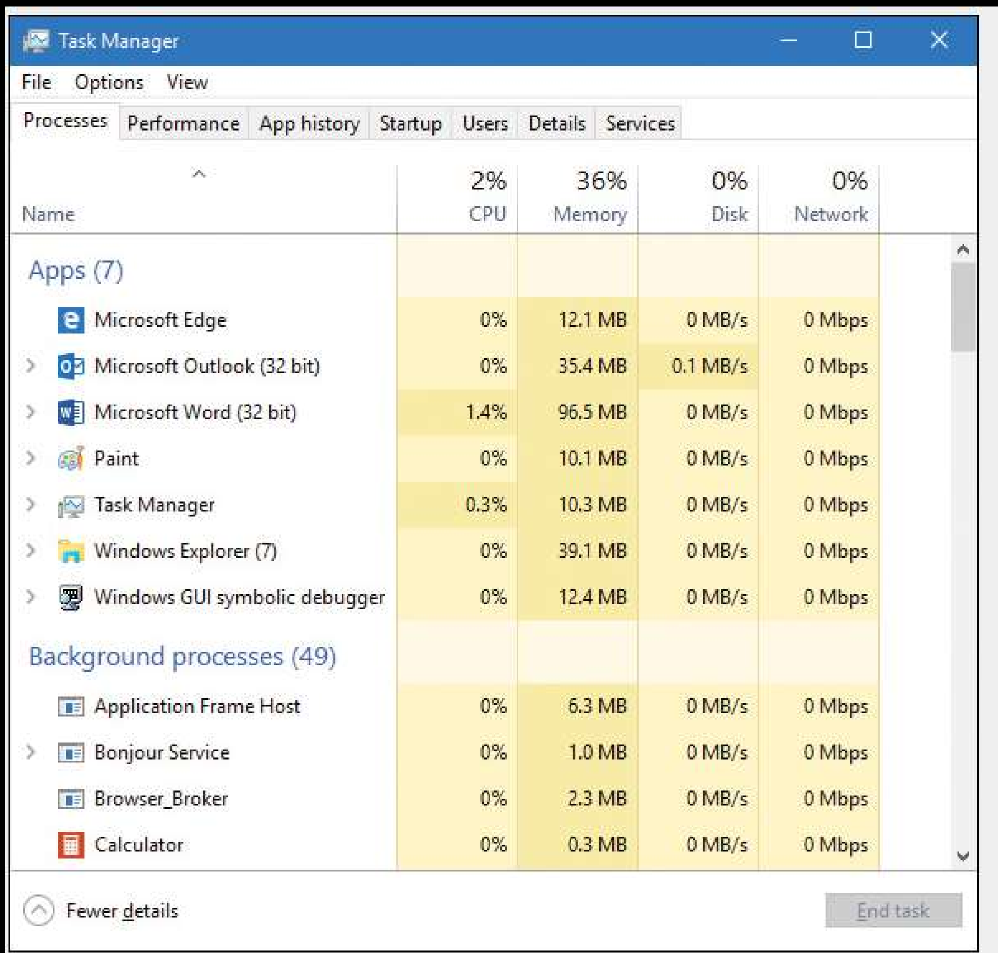
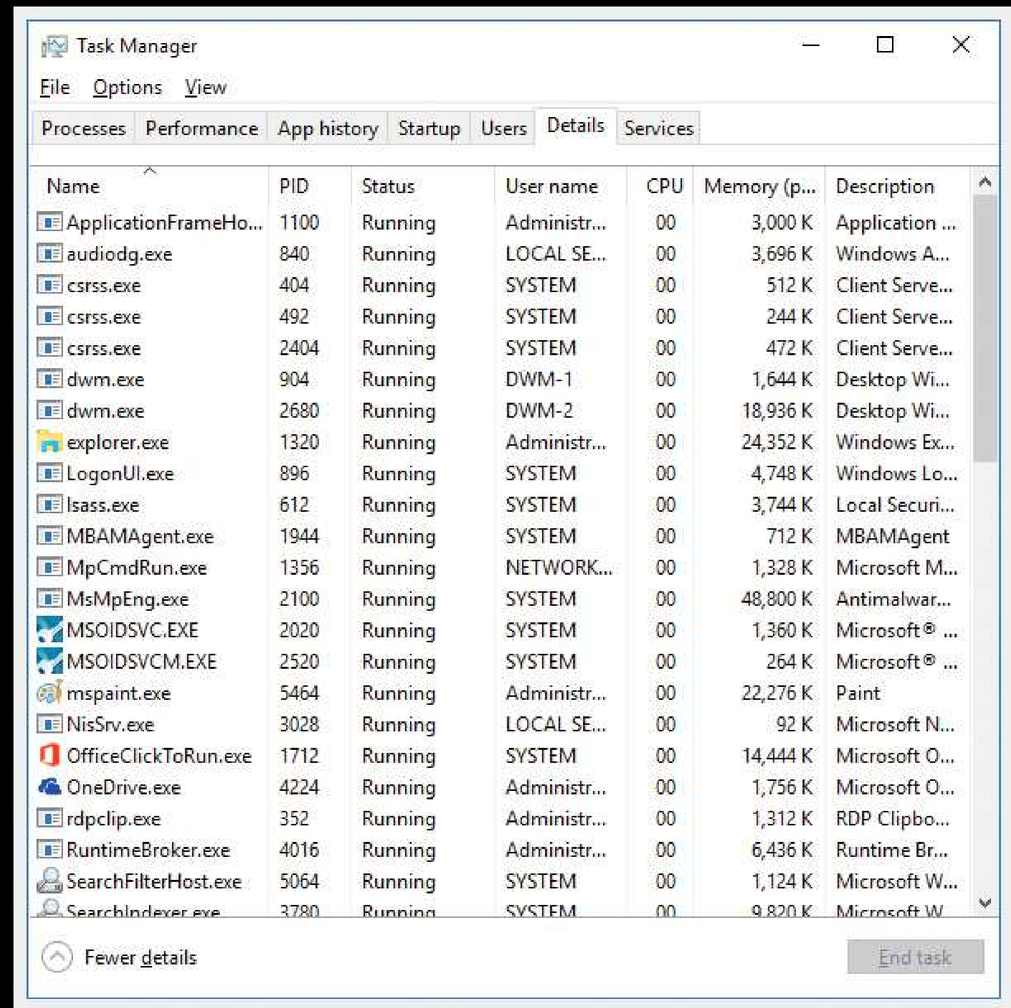
- A program becomes a process when an executable file is loaded into memory



Threads

A thread is a basic unit of CPU utilization; it comprises a thread ID, a program counter, a register set, and a stack. It shares with other threads belonging to the same process its code section, data section, and other operating-system resources, such as open files and signals. A traditional (or heavyweight) process has a single thread of control. If a process has multiple threads of control, it can perform more than one task at a time



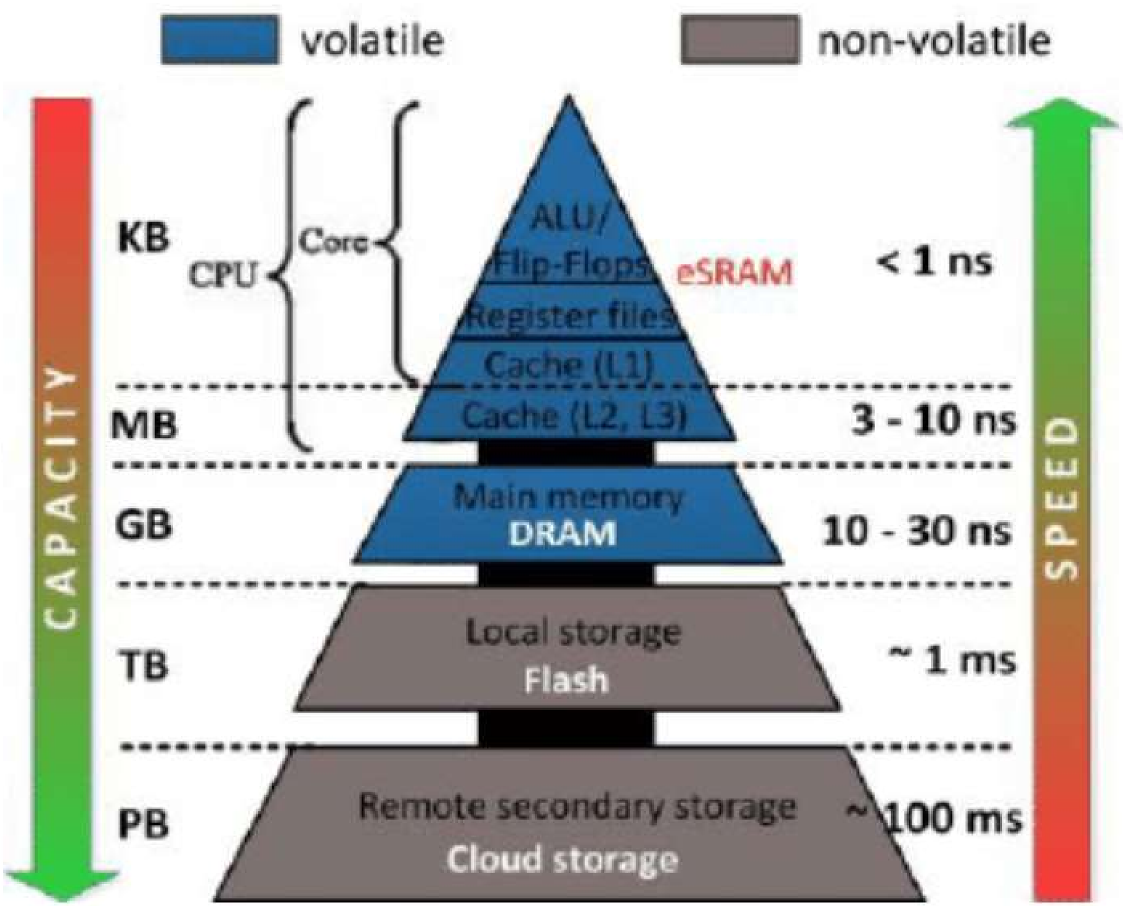

Task Manager
File Options View
Processes Performance App history Startup Users Details Services
2% CPU
36% Memory
0% Disk
0% Network
Name
Apps (7)
Microsoft Edge 0% 12.1 MB 0 MB/s 0 Mbps
Microsoft Outlook (32 bit) 0% 35.4 MB 0.1 MB/s 0 Mbps
Microsoft Word (32 bit) 1.4% 96.5 MB 0 MB/s 0 Mbps
Paint 0% 10.1 MB 0 MB/s 0 Mbps
Task Manager 0.3% 10.3 MB 0 MB/s 0 Mbps
Windows Explorer (7) 0% 39.1 MB 0 MB/s 0 Mbps
Windows GUI symbolic debugger 0% 12.4 MB 0 MB/s 0 Mbps
Background processes (49)
Application Frame Host 0% 6.3 MB 0 MB/s 0 Mbps
Bonjour Service 0% 1.0 MB 0 MB/s 0 Mbps
Browser_Broker 0% 2.3 MB 0 MB/s 0 Mbps
Calculator 0% 0.3 MB 0 MB/s 0 Mbps
Fewer details
End task

Task Manager
File Options View
Processes Performance App history Startup Users Details Services
Name PID Status User name CPU Memory (p... Description
ApplicationFrameHo... 1100 Running Administr... 00 3,000 K Application ...
audiodg.exe 840 Running LOCAL SE... 00 3,696 K Windows A...
csrss.exe 404 Running SYSTEM 00 512 K Client Serve...
csrss.exe 492 Running SYSTEM 00 244 K Client Serve...
csrss.exe 2404 Running SYSTEM 00 472 K Client Serve...
dwm.exe 904 Running DWM-1 00 1,644 K Desktop Wi...
dwm.exe 2680 Running DWM-2 00 18,936 K Desktop Wi...
explorer.exe 1320 Running Administr... 00 24,352 K Windows Ex...
LogonUI.exe 896 Running SYSTEM 00 4,748 K Windows Lo...
lsass.exe 612 Running SYSTEM 00 3,744 K Local Securi...
MBAMAgent.exe 1944 Running SYSTEM 00 712 K MBAMAgent
MpCmdRun.exe 1356 Running NETWORK... 00 1,328 K Microsoft M...
MsMpEng.exe 2100 Running SYSTEM 00 48,800 K Antimalwar...
MSOIDSVC.EXE 2020 Running SYSTEM 00 1,360 K Microsoft® ...
MSOIDSVCM.EXE 2520 Running SYSTEM 00 264 K Microsoft® ...
mspaint.exe 5464 Running Administr... 00 22,276 K Paint
NisSrv.exe 3028 Running LOCAL SE... 00 92 K Microsoft N...
OfficeClickToRun.exe 1712 Running SYSTEM 00 14,444 K Microsoft O...
OneDrive.exe 4224 Running Administr... 00 1,756 K Microsoft O...
rdpclip.exe 352 Running Administr... 00 1,312 K RDP Clipbo...
RuntimeBroker.exe 4016 Running Administr... 00 6,436 K Runtime Br...
SearchFilterHost.exe 5064 Running SYSTEM 00 1,124 K Microsoft W...
Fewer details
End task

```
Tasks: 238 total,   1 running, 184 sleeping,   0 stopped,   0 zombie
%Cpu(s):  7.0 us,  1.3 sy,  0.0 ni, 91.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
KiB Mem :  5939268 total,  1367448 free,  1171108 used,  3400712 buff/cache
KiB Swap:  6801404 total,  6288476 free,   512928 used.  4051952 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
30510 paras     20   0 1238776 201476  78084 S  15.4  3.4   0:26.53 chrome
30591 paras     20   0   41944   3692   3004 R   7.7  0.1   0:00.14 top
 1071 root      20   0  469284 110808  90276 S   2.6  1.9  34:35.39 Xorg
 1324 rabbitmq  20   0 2190040  14520   3164 S   2.6  0.2   7:36.91 beam.smp
 2036 paras     20   0  351068  11348   3800 S   2.6  0.2   0:56.86 ibus-daemon
 2256 paras     20   0 1606948  94192  45184 S   2.6  1.6  36:58.63 compiz
29789 paras     20   0  666292  36848  28652 S   2.6  0.6   0:03.85 gnome-terminal-
    1 root      20   0  185800   4556   2936 S   0.0  0.1   0:03.14 systemd
    2 root      20   0       0      0      0 S   0.0  0.0   0:00.03 kthreadd
    4 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 kworker/0:0H
    6 root       0 -20       0      0      0 I   0.0  0.0   0:00.00 mm_percpu_wq
    7 root      20   0       0      0      0 S   0.0  0.0   0:01.55 ksoftirqd/0
    8 root      20   0       0      0      0 I   0.0  0.0   0:52.59 rcu_sched
    9 root      20   0       0      0      0 I   0.0  0.0   0:00.00 rcu_bh
```

```
top - 20:17:44 up 1 day, 20:59,  1 user,  load average: 0.92, 1.09, 1.31
Tasks: 237 total,   1 running, 183 sleeping,   0 stopped,   0 zombie
%Cpu(s):   0.8/0.2     1[|                                                                     ]
KiB Mem :  5939268 total,  1463788 free,  1093036 used,  3382444 buff/cache
KiB Swap:  6801404 total,  6288476 free,   512928 used.  4129196 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
30956 paras     20   0   41980   3704   3016 R   0.3  0.1   0:01.13 top -u paras
 1856 paras     20   0   45360   2564   2120 S   0.0  0.0   0:00.07 /lib/systemd/systemd --user
 1857 paras     20   0   63880    416      0 S   0.0  0.0   0:00.00 (sd-pam)
 1865 paras     20   0  205300   5340   4728 S   0.0  0.1   0:00.99 /usr/bin/gnome-keyring-daemon --daemonize --login
 1896 paras     20   0   46444   3440   2492 S   0.0  0.1   0:00.65 /sbin/upstart --user
 1988 paras     20   0   32860   1696   1568 S   0.0  0.0   0:00.15 upstart-udev-bridge --daemon --user
 1999 paras     20   0   43968   3944   2612 S   0.0  0.1   0:18.17 dbus-daemon --fork --session --address=unix:abstract=/tmp/dbus-WLnJWhB0Kz
 2011 paras     20   0   86344   3852   3592 S   0.0  0.1   0:00.77 /usr/lib/x86_64-linux-gnu/hud/window-stack-bridge
 2043 paras     20   0  274532   3068   2656 S   0.0  0.1   0:00.14 /usr/lib/gvfs/gvfsd
 2048 paras     20   0  406864   2536   2536 S   0.0  0.0   0:00.00 /usr/lib/gvfs/gvfsd-fuse /run/user/1000/gvfs -f -o big_writes
 2057 paras     20   0  264272   3552   3232 S   0.0  0.1   0:00.02 /usr/lib/ibus/ibus-dconf
 2058 paras     20   0  481844  14316   9072 S   0.0  0.2   0:26.53 /usr/lib/ibus/ibus-ui-gtk3
 2060 paras     20   0  427648   9228   8048 S   0.0  0.2   0:08.17 /usr/lib/ibus/ibus-x11 --kill-daemon
 2080 paras     20   0   32868   1168    968 S   0.0  0.0   0:03.73 upstart-dbus-bridge --daemon --session --user --bus-name session
 2081 paras     20   0   32792    124      0 S   0.0  0.0   0:02.10 upstart-dbus-bridge --daemon --system --user --bus-name system
 2091 paras     20   0  188388   2676   2584 S   0.0  0.0   0:14.66 /usr/lib/ibus/ibus-engine-simple
 2114 paras     20   0   41416   1844   1652 S   0.0  0.0   0:00.05 upstart-file-bridge --daemon --user
 2121 paras     20   0  524848  14400   9612 S   0.0  0.2   0:19.63 /usr/lib/x86_64-linux-gnu/bamf/bamfdaemon
 2122 paras     20   0  166536   2188   2000 S   0.0  0.0   0:00.26 gpg-agent --homedir /home/paras/.gnupg --use-standard-socket --daemon
```

https://www.geeksforgeeks.org/top-command-in-linux-with-examples/

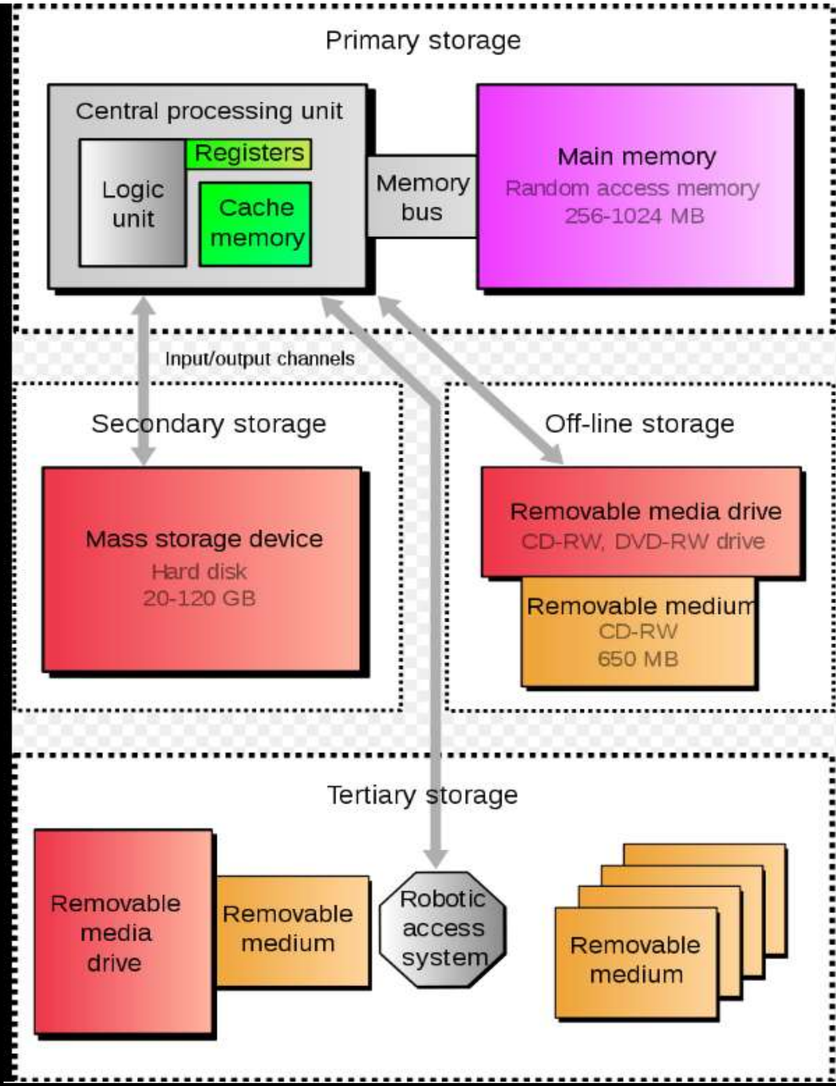
Memory Management



Typical structure of a computer memory hierarchy.

https://www.researchgate.net/figure/Typical-structure-of-a-computer-memory-hierarchy_fig1_281805561

Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80 - 250	25,000 - 50,000	5,000,000
Bandwidth (MB/sec)	20,000 - 100,000	5,000 - 10,000	1,000 - 5,000	500	20 - 150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape



https://en.wikipedia.org/wiki/Computer_data_storage#Primary_storage

Main Memory

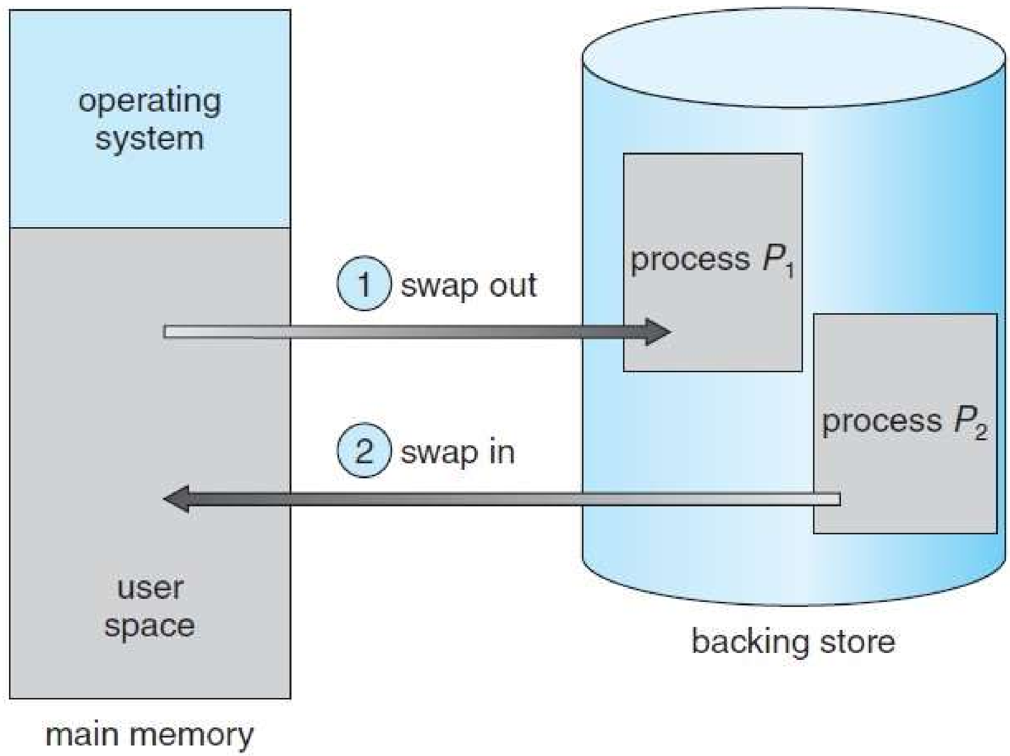
- No memory abstraction – Basic Hardware
- Memory Abstraction: Address spaces
- Virtual Memory

- Main memory and the registers built into the processor itself are the only general-purpose storage that the CPU can access directly.

- There are machine instructions that take memory addresses as arguments, but none that take disk addresses. Therefore, any instructions in execution, and any data being used by the instructions, must be in one of these direct-access storage devices.

- If the data are not in memory, they must be moved there before the CPU can operate on them.

- Swapping
- Contiguous Memory Allocation
- Segmentation
- Paging



Swapping of two processes using a disk as a backing store.

Memory allocation

- First fit
  allocate the first hole that is big enough. Searching can start either at the beginning of the set of holes or at the location where the previous first-fit search ended.We can stop searching as soon as we find a free hole that is large enough

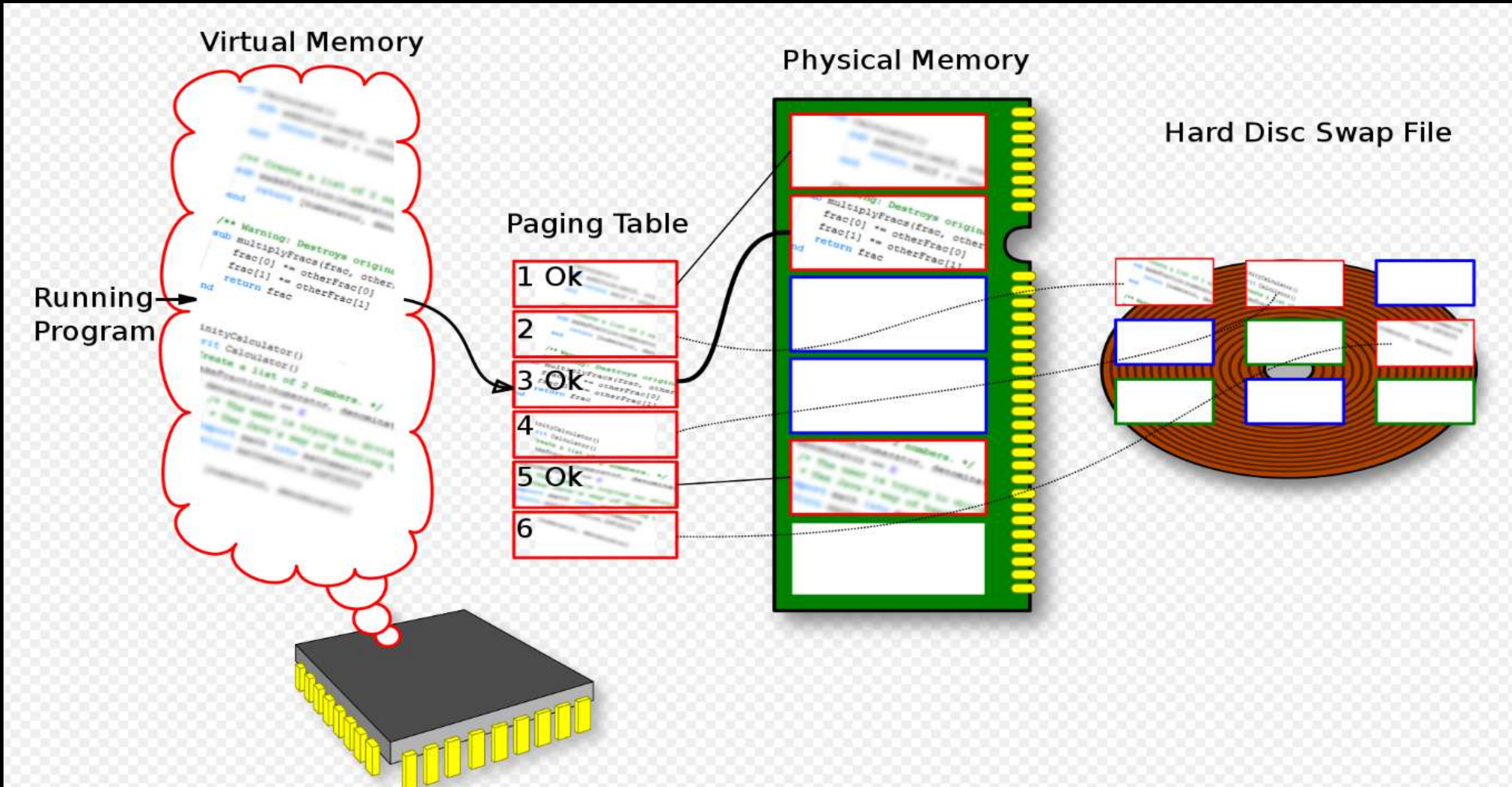
- Best fit. Allocate the smallest hole that is big enough. We must search the entire list, unless the list is ordered by size. This strategy produces the smallest leftover hole.

- Worst fit. Allocate the largest hole. Again, we must search the entire list, unless it is sorted by size. This strategy produces the largest leftover hole, which may be more useful than the smaller leftover hole from a best-fit approach.

Virtual memory

- Virtual memory involves the separation of logical memory as perceived by users from physical memory. This separation allows an extremely large virtual memory to be provided for programmers when only a smaller physical memory is available .

- Virtual memory makes the task of programming much easier, because the programmer no longer needs to worry about the amount of physical memory available; she can concentrate instead on the problem to be programmed



https://en.wikipedia.org/wiki/Memory_address

Part II

- Controlling access to resources
- Authentication
- Operating systems security

Access Control Context

- Authentication: Verification that the credentials of a user or other system entity are valid
- Authorization: The granting of a right or permission to a system entity to access a system resource. This function determines who is trusted for a given purpose
- Audit: An independent review and examination of system records and activities in order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures, to detect breaches in security, and to recommend any indicated changes in control, policy and procedures

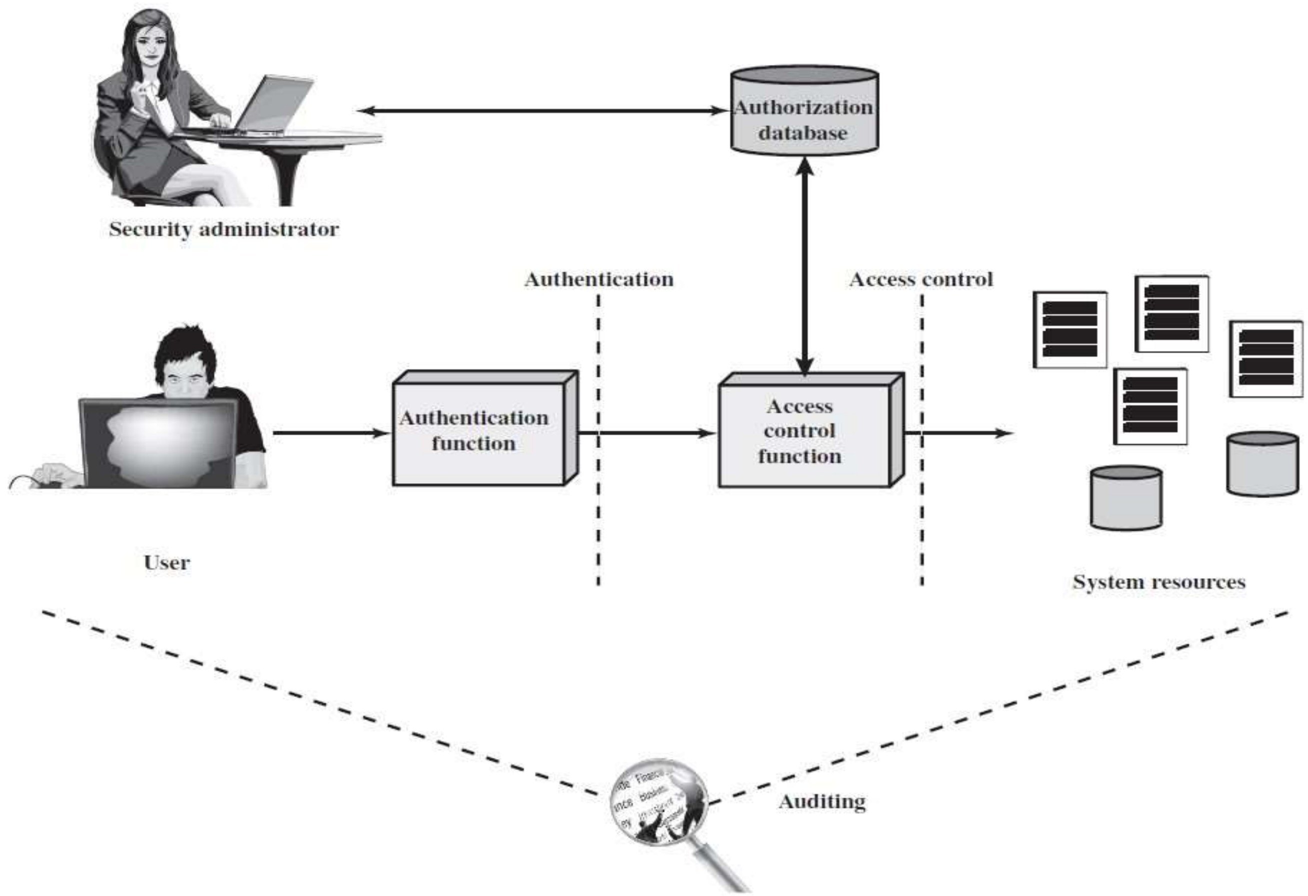


Figure **Relationship Among Access Control and Other Security Functions**
Source: Based on [SAND94].

SAND94 Sandhu, R., and Samarati, P. "Access Control: Principles and Practice." IEEE Communications Magazine, February 1994.

Access Control Policies

- Discretionary access control (DAC): Controls access based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do. This policy is termed discretionary because an entity might have access rights that permit the entity, by its own volition, to enable another entity to access some resource.

- Mandatory access control (MAC): Controls access based on comparing security labels (which indicate how sensitive or critical system resources are) with security clearances (which indicate system entities are eligible to access certain resources). This policy is termed mandatory because an entity that has clearance to access a resource may not, just by its own volition, enable another entity to access that resource.

- Role-based access control (RBAC): Controls access based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles.

- Attribute-based access control (ABAC): Controls access based on attributes of the user, the resource to be accessed, and current environmental conditions.

Subjects, Objects, AND Access rights

A subject is an entity capable of accessing objects. Generally, the concept of subject equates with that of process. Any user or application actually gains access to an object by means of a process that represents that user or application. The process takes on the attributes of the user, such as access rights.

• Owner: This may be the creator of a resource, such as a file. For system resources, ownership may belong to a system administrator. For project resources, a project administrator or leader may be assigned ownership.

• Group: In addition to the privileges assigned to an owner, a named group of users may also be granted access rights, such that membership in the group is sufficient to exercise these access rights. In most schemes, a user may belong to multiple groups.

• World: The least amount of access is granted to users who are able to access the system but are not included in the categories owner and group for this resource.

- An object is a resource to which access is controlled. In general, an object is an entity used to contain and/or receive information.
  - Examples include records, blocks, pages, segments, files, portions of files, directories, directory trees, mailboxes, messages, and programs.
  - Some access control systems also encompass, bits, bytes, words, processors, communication ports, clocks, and network nodes.

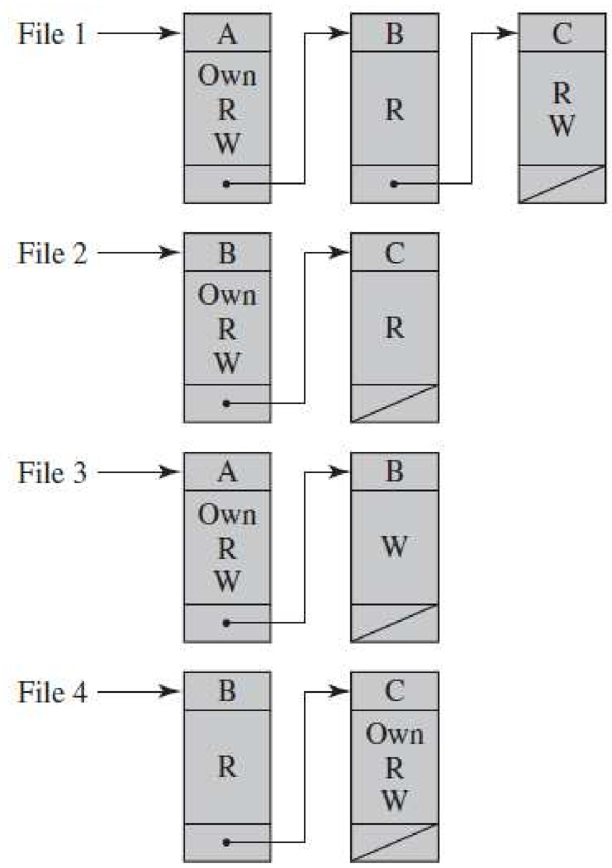
- An access right describes the way in which a subject may access an object
  - Read: User may view information in a system resource (e.g., a file, selected records in a file, selected fields within a record, or some combination). Read access includes the ability to copy or print
  - Write: User may add, modify, or delete data in system resource (e.g., files, records, programs). Write access includes read access
  - Execute: User may execute specified programs
  - Delete: User may delete certain system resources, such as files or records
  - Create: User may create new files, records, or fields
  - Search: User may list the files in a directory or otherwise search the directory

DAC – Discretionary Access Control

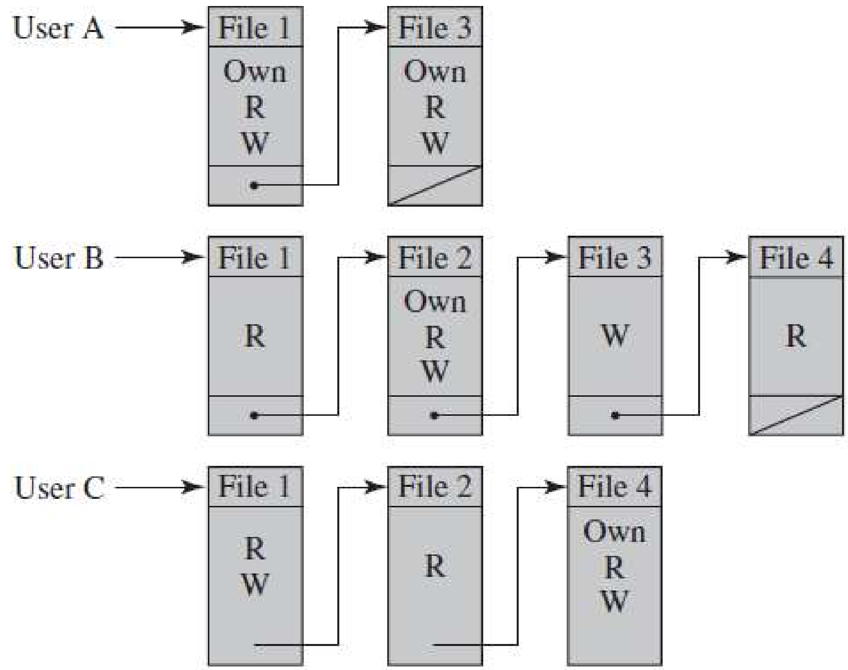
- Access matrix
- Access Control Lists
- Capabilities tickets
- Relational database

		OBJECTS			
		File 1	File 2	File 3	File 4
SUBJECTS	User A	Own Read Write		Own Read Write	
	User B	Read	Own Read Write	Write	Read
	User C	Read Write	Read		Own Read Write

(a) Access matrix



(b) Access control lists for files of part (a)



(c) Capability lists for files of part (a)

Authorization Table for Files

Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

[SAND94] proposes a data structure that is not sparse, like the access matrix, but is more convenient than either ACLs or capability lists. An authorization table contains one row for one access right of one subject to one resource.

Sandhu, R., and Samarati, P. “Access Control: Principles and Practice.” IEEE
Communications Magazine, February 1994.

An Access Control Model

- Processes: Access rights include the ability to delete a process, stop (block), and wake up a process

- Devices: Access rights include the ability to read/write the device, to control its operation (e.g., a disk seek), and to block/unblock the device for use

- Memory locations or regions: Access rights include the ability to read/write certain regions of memory that are protected such that the default is to disallow access

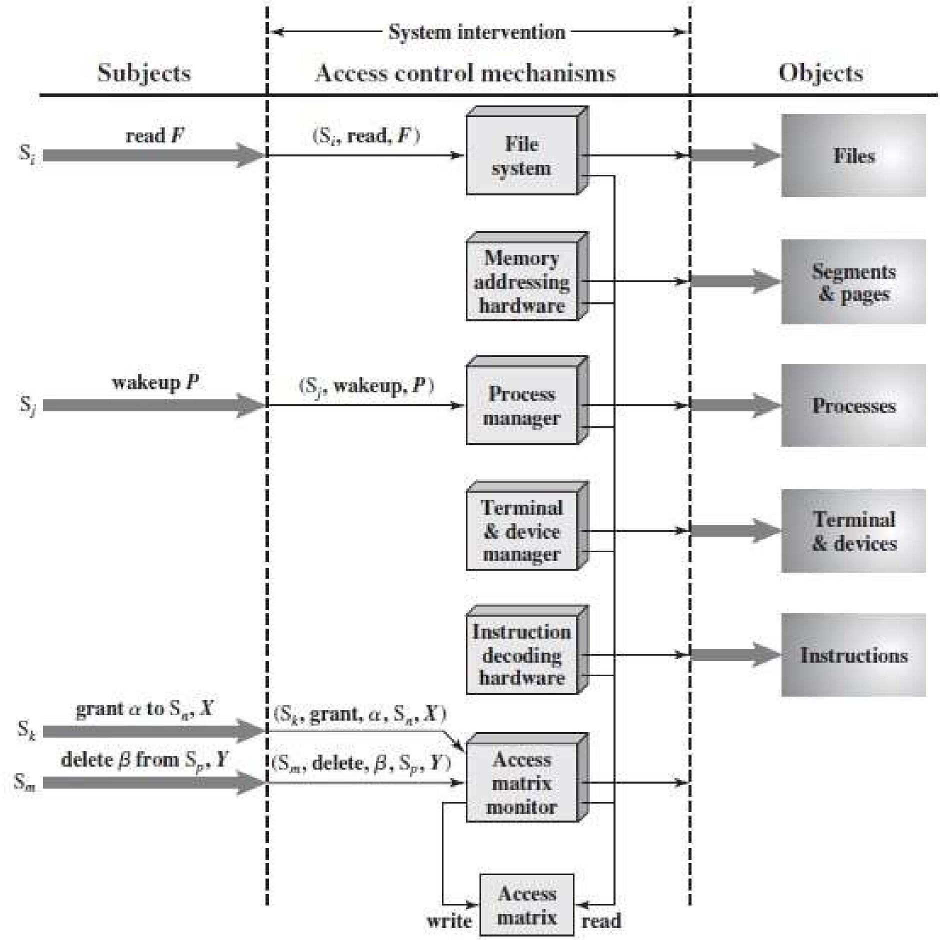
- Subjects: Access rights with respect to a subject have to do with the ability to grant or delete access rights of that subject to other objects, as explained subsequently

OBJECTS

SUBJECTS	Subjects			Files		Processes		Disk drives	
	S_1	S_2	S_3	F_1	F_2	P_1	P_2	D_1	D_2
S_1	control	owner	owner control	read*	read owner	wakeup	wakeup	seek	owner
S_2		control		write*	execute			owner	seek*
S_3			control		write	stop			

* = copy flag set

Extended Access Control Matrix



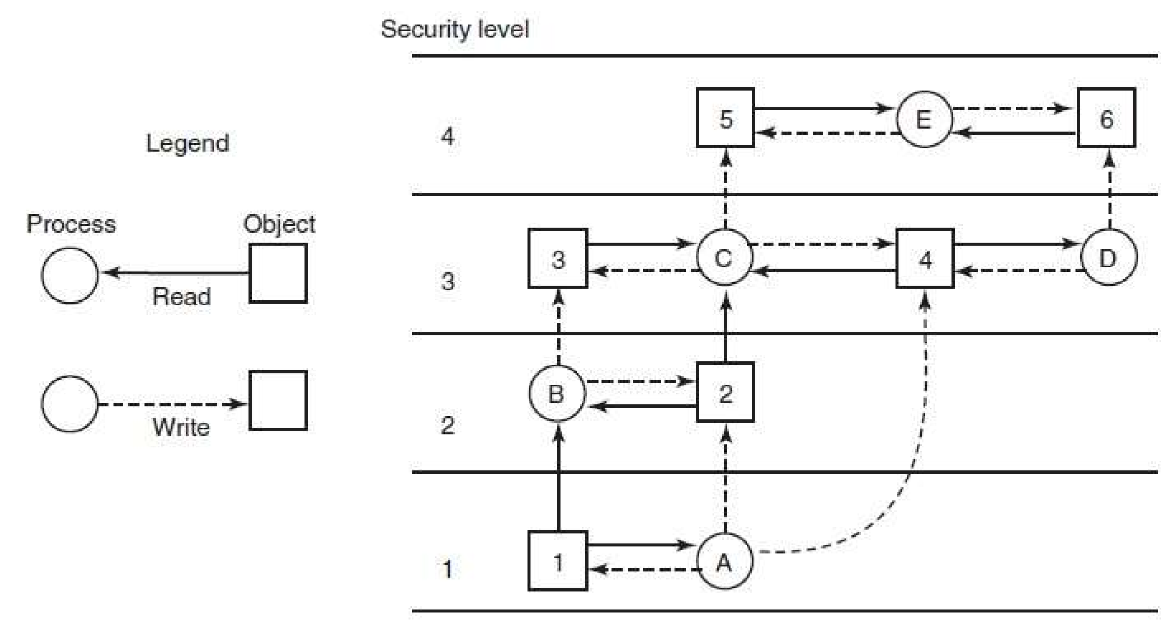
An Organization of the Access Control Function

MAC – Mandatory Access Control
The Bell-LaPadula Model

- Designed for military security – unclassified, confidential, secret and top secret
- The simple security property: A process running at security level k can read only objects at its level or lower. For example, a general can read a lieutenant's documents but a lieutenant cannot read a general'sDocuments
- The * property: A process running at security level k can write only objects at its level or higher. For example, a lieutenant can append a message to a general's mailbox telling everything he knows, but a general cannot append a message to a lieutenant's mailbox telling everything he knows because the general may have seen top-secret documents that may not be disclosed to a lieutenant.

In addition, the BLP model makes a provision for discretionary access control (DAC).

• ds-property: An individual (or role) may grant to another individual (or role) access to a document based on the owner's discretion, constrained by the MAC rules. Thus, a subject can exercise only accesses for which it has the necessary authorization and which satisfy the MAC rules.



The Bell-LaPadula multilevel security model.

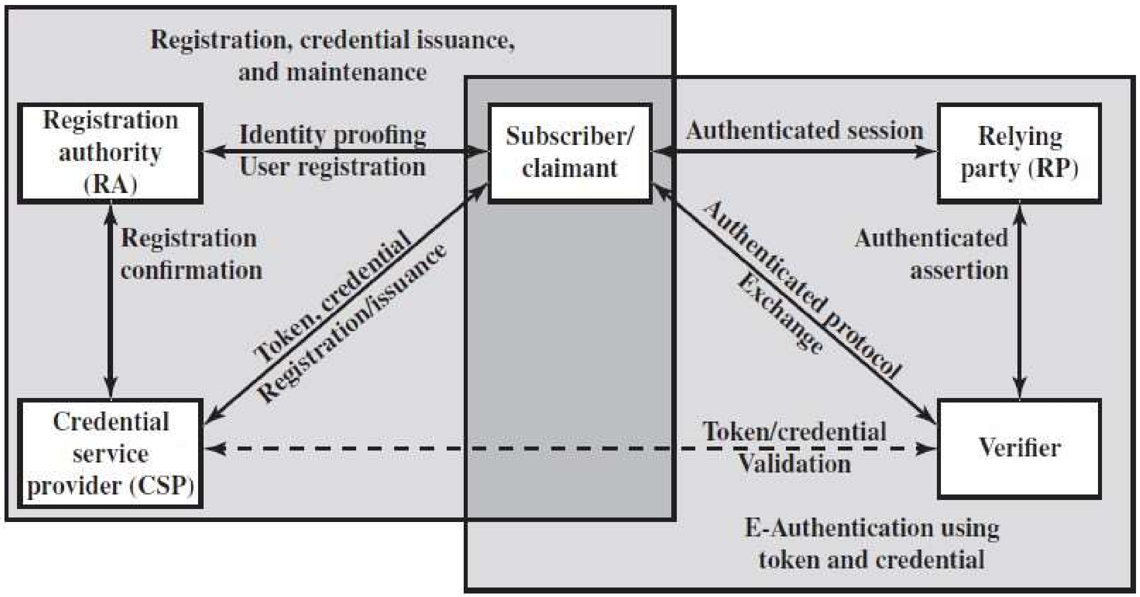
Authentication

• Something the individual knows: Examples includes a password, a personal identification number (PIN), or answers to a prearranged set of questions.

• Something the individual possesses: Examples include electronic keycards, smart cards, and physical keys. This type of authenticator is referred to as a token.

• Something the individual is (static biometrics): Examples include recognition by fingerprint, retina, and face.

• Something the individual does (dynamic biometrics): Examples include recognition by voice pattern, handwriting characteristics, and typing rhythm.



The NIST SP 800-63-2 E-Authentication Architectural Model

Operating Systems Security

System Security Planning

- The purpose of the system, the type of information stored, the applications and services provided, and their security requirements

- The categories of users of the system, the privileges they have, and the types of information they can access

- How the users are authenticated

- How access to the information stored on the system is managed

- What access the system has to information stored on other hosts, such as file or database servers, and how this is managed.

- Who will administer the system, and how they will manage the system (via local or remote access)

- Any additional security measures required on the system, including the use of host firewalls, anti-virus or other malware protection mechanisms, and logging

Operating Systems Hardening

- Install and patch the operating system
- Harden and configure the operating system to adequately address the identified security needs of the system by:
  - Removing unnecessary services, applications, and protocols
  - Configuring users, groups, and permissions
  - Configuring resource controls
- Install and configure additional security controls, such as anti-virus, hostbased firewalls, and intrusion detection systems (IDS), if needed
- Test the security of the basic operating system to ensure that the steps taken adequately address its security needs

Part III - Lab

- Buffer Overflow Attacks - TBD

- Insider attacks - TBD

- Malware - TBD

Bibliography

- Andrew S. Tanenbaum – Modern Operating System – 4th Edition
- Abraham Silberschatz – Operating Systems Concepts – 9th Edition
- William Stallings – Computer Security Principles and Practice – 3rd Edition