ЗАДАЧІ ТА ІНСТРУМЕНТИ ОПЕРАЦІЙНИХ СИСТЕМ

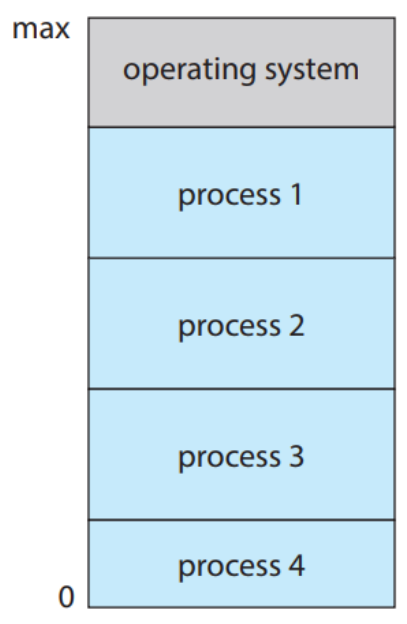
Питання 1.3.

ОС забезпечує середовище, в якому виконуються програми

- Всередині різні ОС побудовані по-різному, проте існує багато спільних компонентів.
- Запуск комп'ютера управляється завантажувачем ОС (bootstrap program), який зазвичай зберігається у вбудованій пам'яті комп'ютера (firmware).
  - Ініціалізує всі аспекти системи, від регістрів ЦП до контролерів пристроїв до вмісту пам'яті.
  - Для завантаження ОС завантажувачу потрібно знати, де в пам'яті знаходиться ядро ОС.
  - Як тільки ядро завантажене та виконується, воно може постачати служби для системи та користувачів.
  - Деякі служби постачаються ззовні ядра системними програмами, які завантажуються в пам'ять під час boot time та стають ***системними демонами (system daemons)***, працюючи весь час роботи ядра.
  - На Linux перша системна програма – "systemd," вона запускає багато інших демонів.
- Якщо немає процесів для виконання, пристроїв вводу-виводу для обслуговування та користувачів, з якими потрібно взаємодіяти, ОС очікуватиме на події.
  - Майже завжди про подію сигналізує переривання або виняток (trap, exception).

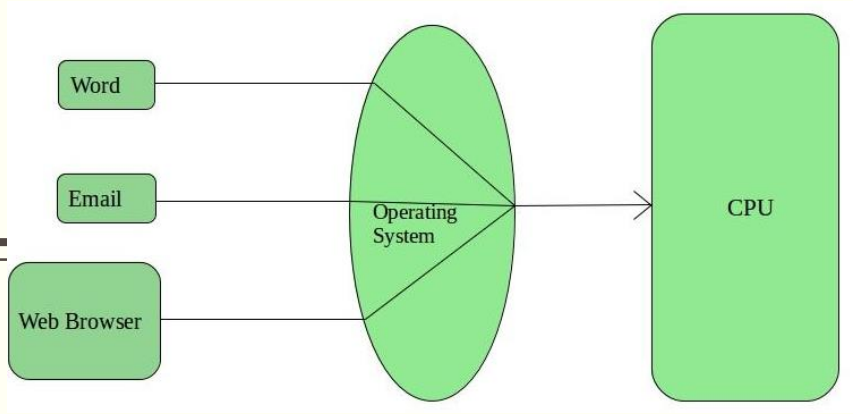
Мультипрограмування

- Користувачі бажають запускати багато програм одночасно.
  - Мультипрограмування підвищує CPU utilization та організує виконання програм так, щоб ЦП завжди виконував одну з них.
  - У мультипрограмній системі програма в роботі називається ***процесом***.



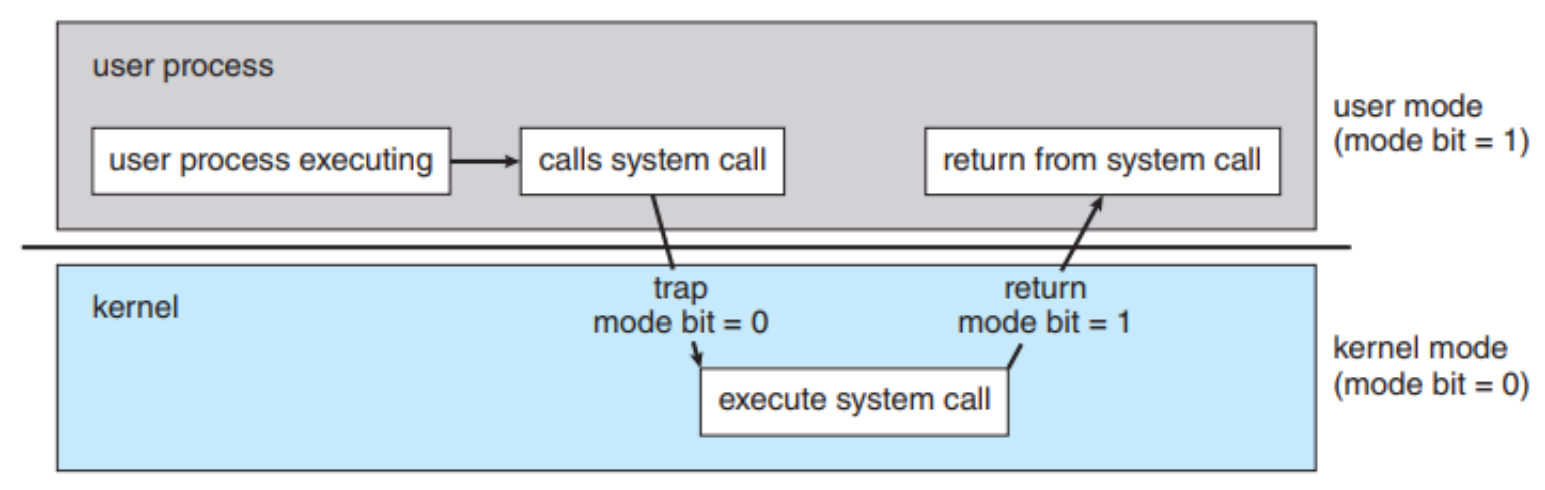
- Ідея: ОС тримає кілька процесів у пам'яті одночасно.
  - ОС обирає та починає виконувати один з процесів.
  - У певний момент процесу потрібно очікувати завершення деякого завдання, наприклад, операції вводу-виводу.
  - У мультипрограмних системах ОС перемикається на інший процес та виконує його. І т. д.
  - З часом перший процес завершує очікування та отримує ЦП-ресурс назад.
  - As long as at least one process needs to execute, the CPU is never idle.

Багатозадачність



- Логічне розширення мультипрограмування.
  - ЦП виконує багато процесів, перемикаючись між ними, проте перемикання (switching) відбуваються часто, забезпечуючи високу швидкість реакції.
  - Ввід-вивід може бути інтерактивним, проте з «швидкістю людини»: вивід на дисплей, а ввід з клавіатури, миші чи сенсорного екрану.
  - ОС швидко перемикатиме ЦП на інший процес, поки буде відбуватись ввід даних.
- Потрібне управління пам’яттю для розміщення в ній кількох процесів.
  - Якщо кілька процесів готові до запуску в один момент часу, ОС вирішує, який запускати – CPU scheduling.
  - Також необхідно обмежувати вплив процесів один на одного на всіх етапах роботи ОС – планування процесів, робота з дисками, управління пам’яттю та ін.
- Багатозадачна ОС повинна забезпечувати reasonable response time.
  - Поширений спосіб – віртуальна пам’ять – техніка, яка дозволяє виконувати процес, що не повністю розташований в пам’яті.
  - Основна перевага: запуск програм, які перевищують за об’ємом фізичну пам’ять пристрою.
  - Віртуальна пам’ять абстрагує основну пам’ять у великий, рівномірний масив-сховище, відокремлюючи логічну пам’ять (з точки зору користувача) від фізичної пам’яті.

Dual-Mode and Multimode Operation



- Для забезпечення захищеності ОС необхідно розділяти виконання коду ОС та користувацького коду (програм).
  - Виокремимо 2 режими операцій: ***user mode*** та ***kernel mode*** (supervisor mode, system mode, privileged mode).
  - Спеціальний mode-біт додається до апаратного забезпечення комп'ютера, щоб визначати поточний режим: kernel (0) or user (1).
  - Коли комп'ютерна система виконує користувацький додаток, система знаходиться в режимі користувача.
  - Коли додаток запитує службу від ОС через системний виклик, система повинна перейти в режим ядра.

Процес завантаження ОС

- Протягом завантаження ОС апаратне забезпечення запускається в режимі ядра.
  - Потім ОС завантажується та запускає користувацькі додатки в режимі користувача.
  - Як тільки трапляться виняток або переривання, апаратне забезпечення перемикається з режиму користувача в режим ядра (mode-біт у 0).
  - Thus, whenever the operating system gains control of the computer, it is in kernel mode.
  - The system always switches to user mode (by setting the mode bit to 1) before passing control to a user program.
- Подвійний (dual) режим операцій забезпечує захист ОС від errant users—і errant users одних від одного.
  - Машинні інструкції, які можуть нанести шкоду, позначаються привілейованими.
  - Апаратне забезпечення дозволяє привілейованим інструкціям виконуватись тільки в режимі ядра.
  - Спроба виконання привілейованої інструкції в режимі користувача призведе у більшості випадків до винятку.
- Інструкція щодо перемикання в режим ядра – приклад привілейованої інструкції.
  - Інші приклади: управління вводом-виводом, таймером, перериваннями та ін.

Розширення концепції режимів

- Процесори Intel мають 4 окремих кільця захисту:
  - ring 0 = kernel mode, ring 3 = user mode. Решта кілець (для служб ОС) на практиці використовуються рідко.
- Системи на ARMv8 мають 7 режимів.
  - ЦП з підтримкою віртуалізації мають окремий режим, що вказує на управління системою менеджером віртуальної машини (VMM).
  - У цьому режимі VMM має більше привілеїв, ніж користувач, проте менше, ніж ядро. Потрібно для створення та управління віртуальними машинами, зміни стану ЦП тощо.
- Системні виклики забезпечують можливість запиту користувацькою програмою до ОС з метою виконання tasks reserved for the operating system on the user program's behalf.
  - Звернення до системних викликів може відбуватись по-різному залежно від процесора.
  - Системний виклик зазвичай відбувається у формі trap на конкретне місце вектору переривань.
  - Цей виняток (trap) може виконуватись by a generic trap instruction, хоч деякі системи мають специфічну інструкцію syscall, щоб звертатись до системних викликів.

- Коли виконується системний виклик, він зазвичай обробляється залізом як software interrupt.
  - Управління передається через вектор переривань до службової підпрограми ОС, а mode bit = 0.
  - Ядро переглядає перериваючу інструкцію, щоб визначити, який системний виклик трапився; параметр визначає тип служби, яку запитує user program.
  - Додаткова інформація для запиту може передаватись у регістри, через стек або в пам’ять (вказівники на комірки пам’яті passed in registers).
  - Ядро перевіряє коректність параметрів, виконує запит та поверне управління наступній після системного виклику інструкції.
- Once hardware protection is in place, it detects errors that violate modes.
  - These errors are normally handled by the operating system.
  - If a user program fails in some way—such as by making an attempt either to execute an illegal instruction or to access memory that is not in the user’s address space—then the hardware traps to the operating system.
  - The trap transfers control through the interrupt vector to the operating system, just as an interrupt does.
  - When a program error occurs, the operating system must terminate the program abnormally.
  - This situation is handled by the same code as a user-requested abnormal termination.
  - An appropriate error message is given, and the memory of the program may be dumped.
  - The memory dump is usually written to a file so that the user or programmer can examine it and perhaps correct it and restart the program.

Таймер

- We cannot allow a user program to get stuck in an infinite loop or to fail to call system services and never return control to the operating system.
  - To accomplish this goal, we can use a timer.
  - A timer can be set to interrupt the computer after a specified period.
  - The period may be fixed (for example, 1/60 second) or variable (for example, from 1 millisecond to 1 second).
- A variable timer is generally implemented by a fixed-rate clock and a counter.
  - The operating system sets the counter.
  - Every time the clock ticks, the counter is decremented.
  - When the counter reaches 0, an interrupt occurs.
  - For instance, a 10-bit counter with a 1-millisecond clock allows interrupts at intervals from 1 millisecond to 1,024 milliseconds, in steps of 1 millisecond.
- Before turning over control to the user, the operating system ensures that the timer is set to interrupt.
  - If the timer interrupts, control transfers automatically to the operating system, which may treat the interrupt as a fatal error or may give the program more time.
  - Clearly, instructions that modify the content of the timer are privileged.

Управління процесами

- Програма не може щось робити, поки її інструкції не будуть виконані ЦП.
  - Програма на зразок компілятора – це процес, текстовий редактор, запущений окремим користувачем на ПК – це процес. social media app, запущений на мобільному пристрої – це процес.
  - Доступна можливість забезпечити системні виклики, які дозволять процесам створювати підпроцеси (subprocesses) для їх конкурентного виконання.
- Процесу потрібні ресурси для виконання завдання (task): CPU time, memory, files, and I/O devices тощо.
  - Крім різноманітних фізичних та логічних ресурсів, виділених при створенні процесу, можуть передаватись різні дані для ініціалізації – вхідні дані.
  - Наприклад, для працюючого браузеру потрібне URL-посилання, щоб відобразити веб-сторінку.
  - Коли процес переривається (terminate), ОС will reclaim any reusable resources.

Управління процесами

- Програма сама по собі не є процесом.
  - Це пасивна сутність, як вміст файлу на диску, а процес – активна сутність.
  - Однопоточний процес має one program counter, який послідовно визначає наступну для виконання інструкцію.
  - У будь-який момент часу виконується максимум 1 інструкція такого процесу.
  - Хоч 2 процеси можуть бути пов’язаними з однією програмою, вони розглядаються як 2 окремих послідовності виконання.
  - Багатопоточний процес має багато program counters, кожен з яких вказує на наступну інструкцію для виконання даним потоком.
- Процес – одиниця роботи в системі.
  - A system consists of a collection of processes, some of which are operating-system processes (those that execute system code) and the rest of which are user processes (those that execute user code).
  - All these processes can potentially execute concurrently—by multiplexing on a single CPU core—or in parallel across multiple CPU cores.

Управління процесами

- ОС відповідає за наступні дії в контексті управління процесами:
  - Створення та видалення як користувацьких, так і системних процесів
  - Планування процесів та потоків на рівні ЦП
  - Зупинка та відновлення роботи процесів
  - Забезпечення механізмів синхронізації процесів
  - Забезпечення механізмів комунікації процесів

Управління пам’яттю

- For a program to be executed, it must be mapped to absolute addresses and loaded into memory.
  - As the program executes, it accesses program instructions and data from memory by generating these absolute addresses.
  - Eventually, the program terminates, its memory space is declared available, and the next program can be loaded and executed.
- To improve both the utilization of the CPU and the speed of the computer’s response to its users, general-purpose computers must keep several programs in memory, creating a need for memory management.
  - Many different memory-management schemes are used.
  - These schemes reflect various approaches, and the effectiveness of any given algorithm depends on the situation.
  - In selecting a memory-management scheme for a specific system, we must take into account many factors—especially the hardware design of the system.
  - Each algorithm requires its own hardware support.

Управління пам’яттю

- ОС відповідає за наступні дії щодо управління пам’яті:
  - Відстеження, які частини пам’яті зараз використовуються та які процеси їх використовують
  - Виділення та вивільнення пам’яті за потреби
  - Обирає, які процеси (їх частини) та дані переміщати в / з пам’яті

Управління файловими системами

- ОС постачає єдине, логічне представлення of information storage.
  - ОС абстрагує від фізичних властивостей пристроїв зберігання, щоб визначати логічну одиницю зберігання – файл.
- Файл – набір пов’язаної інформації, визначеної її creator-ом.
  - Файли представляють програми (у вигляді первинного чи об’єктного коду) та дані.
  - Файли даних можуть містити numeric, alphabetic, alphanumeric або binary дані.
  - Files may be freeform (for example, text files), or they may be formatted rigidly (for example, fixed fields such as an mp3 music file).
  - Clearly, the concept of a file is an extremely general one.
- ОС реалізує концепцію файлу, управляючи mass storage media та пристроями, що ним управляють.
  - Також файли зазвичай групуються в папки для спрощення користування.
  - При багатокористувацькому доступі до файлів бажано управляти доступом окремих користувачів до них (зчитування, запис, дозапис тощо).
- ОС відповідає за наступні дії в контексті управління файлами:
  - Creating and deleting files
  - Creating and deleting directories to organize files
  - Підтримує програмні примітиви для роботи з файлами та папками
  - Відображає файли на mass storage
  - Здійснює резервне копіювання файлів у стабільному (nonvolatile) середовищі зберігання

Управління сховищами даних (Mass-Storage Management)

- Most programs—including compilers, web browsers, word processors, and games—are stored on these devices until loaded into memory.
  - The programs then use the devices as both the source and the destination of their processing.
  - Hence, the proper management of secondary storage is of central importance to a computer system.
- The operating system is responsible for the following activities in connection with secondary storage management:
  - Mounting and unmounting
  - Free-space management
  - Storage allocation
  - Disk scheduling
  - Partitioning
  - Protection

Управління сховищами даних (Mass-Storage Management)

- Because secondary storage is used frequently and extensively, it must be used efficiently.
  - The entire speed of operation of a computer may hinge on the speeds of the secondary storage subsystem and the algorithms that manipulate that subsystem.
- At the same time, there are many uses for storage that is slower and lower in cost (and sometimes higher in capacity) than secondary storage.
  - Backups of disk data, storage of seldom-used data, and long-term archival storage are some examples.
  - Magnetic tape drives and their tapes and CD DVD and Blu-ray drives and platters are typical tertiary storage devices.
- Tertiary storage is not crucial to system performance, but it still must be managed.
  - Some operating systems take on this task, while others leave tertiary-storage management to application programs.
  - Some of the functions that operating systems can provide include mounting and unmounting media in devices, allocating and freeing the devices for exclusive use by processes, and migrating data from secondary to tertiary storage.

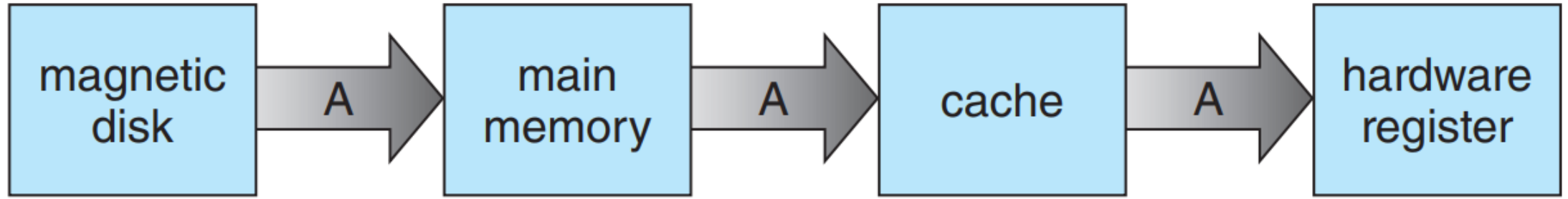
Управління кешем

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

- Information is normally kept in some storage system (such as main memory).
  - As it is used, it is copied into a faster storage system—the cache—on a temporary basis.
  - When we need a particular piece of information, we first check whether it is in the cache.
  - If it is, we use the information directly from the cache.
  - If it is not, we use the information from the source, putting a copy in the cache under the assumption that we will need it again soon.

- In addition, internal programmable registers provide a high-speed cache for main memory.
  - The programmer (or compiler) implements the register-allocation and register-replacement algorithms to decide which information to keep in registers and which to keep in main memory.
- Other caches are implemented totally in hardware.
  - For instance, most systems have an instruction cache to hold the instructions expected to be executed next.
  - Without this cache, the CPU would have to wait several cycles while an instruction was fetched from main memory.
  - For similar reasons, most systems have one or more high-speed data caches in the memory hierarchy.
  - We are not concerned with these hardware-only caches in this text, since they are outside the control of the operating system.
- Because caches have limited size, cache management is an important design problem.
  - Careful selection of the cache size and of a replacement policy can result in greatly increased performance.

Migration of integer A from disk to register



- Переміщення інформації між рівнями ієрархії пам’яті може бути явним чи неявним, залежно від заліза та управляючого ПЗ ОС.
  - Наприклад, передача даних з кешу в ЦП та регістри зазвичай є хардверною функцією, без втручання ОС.
  - Навпаки, передача даних з диску в пам’ять зазвичай управляється ОС.
- В ієрархічній структурі пам’яті ті ж дані можуть з’являтись на різних рівнях системи зберігання інформації.
  - Наприклад, нехай ціле число A, яке буде інкрементуватись на 1, знаходиться в файлі B, (жорсткий диск).
  - Операція інкременту виконується так: (1) операція вводу-виводу для копіювання A в основну пам’ять; (2) копіювання A в кеш та внутрішній регістр; (3) виконання операції процесором …

- In a computing environment where only one process executes at a time, this arrangement poses no difficulties, since an access to integer A will always be to the copy at the highest level of the hierarchy.
  - However, in a multitasking environment, where the CPU is switched back and forth among various processes, extreme care must be taken to ensure that, if several processes wish to access A, then each of these processes will obtain the most recently updated value of A.
- The situation becomes more complicated in a multiprocessor environment where, in addition to maintaining internal registers, each of the CPUs also contains a local cache.
  - In such an environment, a copy of A may exist simultaneously in several caches.
  - Since the various CPUs can all execute in parallel, we must make sure that an update to the value of A in one cache is immediately reflected in all other caches where A resides.
  - This situation is called cache coherency, and it is usually a hardware issue (handled below the operating-system level).
- In a distributed environment, the situation becomes even more complex.
  - In this environment, several copies (or replicas) of the same file can be kept on different computers.
  - Since the various replicas may be accessed and updated concurrently, some distributed systems ensure that, when a replica is updated in one place, all other replicas are brought up to date as soon as possible.

Управління системою вводу-виводу

- One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from the user.
  - For example, in UNIX, the peculiarities of I/O devices are hidden from the bulk of the operating system itself by the I/O subsystem.
- The I/O subsystem consists of several components:
  - A memory-management component that includes buffering, caching, and spooling
  - A general device-driver interface
  - Drivers for specific hardware devices
- Only the device driver knows the peculiarities of the specific device to which it is assigned.

Безпека (security) та захист (protection) ОС

- Якщо комп’ютер має кілька користувачів та дозволяє конкурентне виконання кількох процесів, доступ до даних потрібно регулювати.
  - ОС впроваджує механізми, які забезпечують роботу з ресурсами (файлами, сегментами пам’яті, ЦП та ін.) тільки тих процесів, які авторизували свій доступ.
  - Наприклад, memory-addressing hardware забезпечує можливість виконання процесу лише всередині свого адресного простору.
  - Таймер керує тим, щоб жоден процес не міг отримати доступ до ЦП без eventually relinquishing control.
  - Device-control registers недоступні для користувачів, тому захищена цілісність периферійних пристроїв..
- Захист, як і будь-який механізм контролю доступу процесів або користувачів до ресурсів, визначається комп’ютерною системою.
  - This mechanism must provide means to specify the controls to be imposed and to enforce the controls.
- Захист може покращити надійність шляхом відстеження прихованих помилок at the interfaces between component subsystems.
  - Early detection of interface errors can often prevent contamination of a healthy subsystem by another subsystem that is malfunctioning.
  - Furthermore, an unprotected resource cannot defend against use (or misuse) by an unauthorized or incompetent user.

- Система може мати адекватний захист, проте схильною до відмов та надавати недоречний доступ.
  - Нехай існує користувач, чиї дані аутенифікації були викрадені.
  - Дані такого користувача можна копіювати чи видаляти, незважаючи на працюючий захист пам’яті.
  - Безпека системи передбачає захист від зовнішніх та внутрішніх атак: вірусів, черв’яків, denial-of-service attacks (which use all of a system’s resources and so keep legitimate users out of the system), identity theft, and theft of service (unauthorized use of a system). Prevention of some of these attacks is considered an operating system function on some systems, while other systems leave it to policy or additional software. Due to the alarming rise in security incidents, operatingsystem security features are a fast-growing area of research and implementation.
- Protection and security require the system to be able to distinguish among all its users. Most operating systems maintain a list of user names and associated user identifier (user IDs). In Windows parlance, this is a security ID (SID). These numerical IDs are unique, one per user. When a user logs in to the system, the authentication stage determines the appropriate user ID for the user. That user ID is associated with all of the user’s processes and threads. When an ID needs to be readable by a user, it is translated back to the user name via the user name list.

- In some circumstances, we wish to distinguish among sets of users rather than individual users.
  - For example, the owner of a file on a UNIX system may be allowed to issue all operations on that file, whereas a selected set of users may be allowed only to read the file.
  - To accomplish this, we need to define a group name and the set of users belonging to that group.
  - Group functionality can be implemented as a system-wide list of group names and group identifier.
- A user can be in one or more groups, depending on operating-system design decisions.
  - The user's group IDs are also included in every associated process and thread.
- In the course of normal system use, the user ID and group ID for a user are sufficient.
  - However, a user sometimes needs to escalate privileges to gain extra permissions for an activity.
  - The user may need access to a device that is restricted, for example.
  - Operating systems provide various methods to allow privilege escalation.
  - On UNIX, for instance, the setuid attribute on a program causes that program to run with the user ID of the owner of the file, rather than the current user's ID.
  - The process runs with this effective UID until it turns off the extra privileges or terminates.