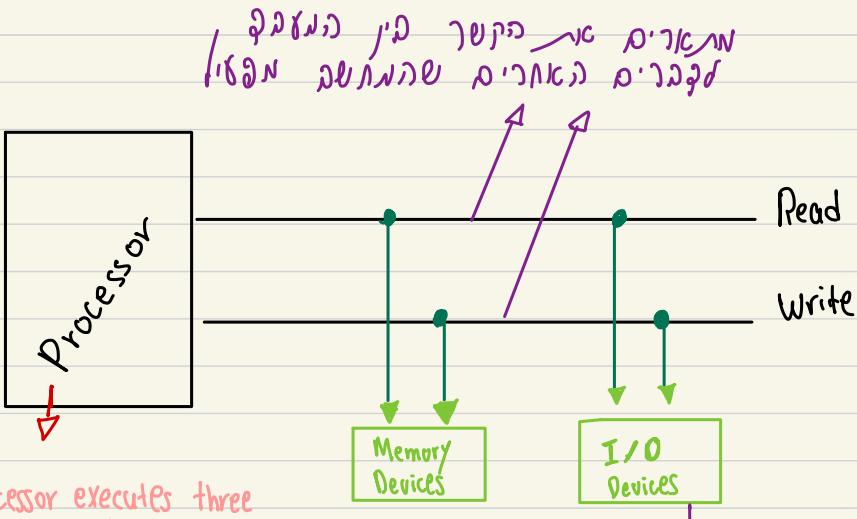


Memory Mapped I/O

processor : provides the instructions and processing power the computer needs to do its work.

Memory device : Is a piece of hardware used to store data

I/O Devices : Is any hardware used by a human operator or other system to communicate with a computer.



processor executes three basic functions :

- ① receive input data
- ② process this data
- ③ provide output data.

.. CN, CGIO, CPIO
.. PCI, VIDE CARD
.. SCSI, IDE
.. GPNIC, GND

Continue with the processor

* processor contains several important features:

- ① **Socket:** Is a type of physical connection between the processor and the motherboard and it's responsible for transmitting energy to the processor.
- ② **Core:** Are responsible for the processing speed, the more core processor has the more functions it can execute simultaneously without overloading your system.
- ③ **Clock:** Is responsible for defining the frequency that the processor will use to execute one task, It's measured by Hertz. this mean the amount of cycles that can happen in a given amount of time in this case in seconds.
- ④ **Cache:** Is the auxiliary memory of a processor, where the most accessed data that will be processed is identified and then stored, the processor accessed this memory and is able to execute the action faster because it's more readily available.

- Modern:

Northbridge

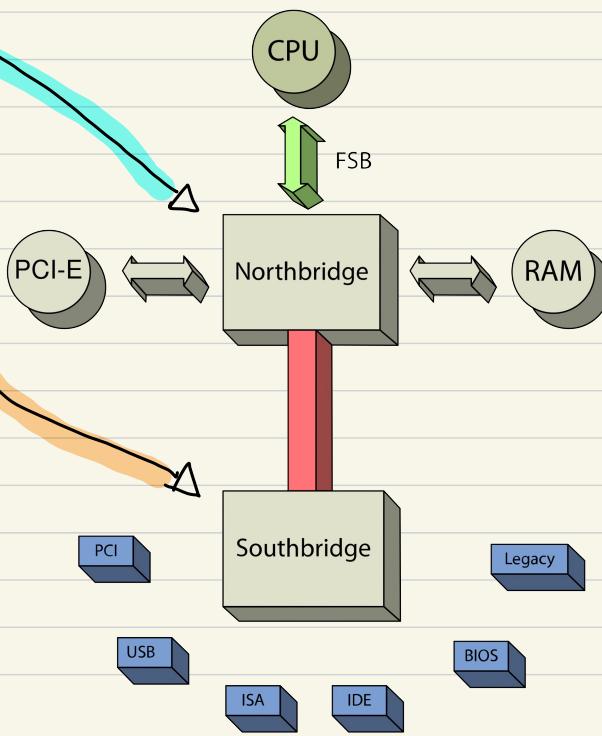
: It's directly connected to the CPU, RAM, AGP...

If it's required for the CPU to communicate with AGP or PCI. Northbridge operates at a faster speed as it connects to high-speed components in the computer.

Southbridge

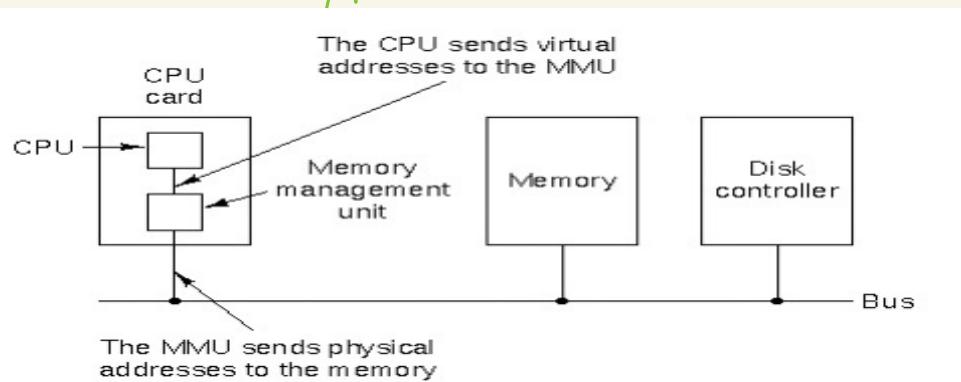
: It connects the components such as Bios, SATA, IDE connectors and USB ports.

If need to communicate with the CPU, the information has to go through the southbridge, then northbridge and finally to the CPU



Memory:

- **RAM**: Random Access Memory - is the hardware in a computing device where the operating system, Application programs and data in current use are kept so they can be quickly reached by the device's processor. RAM is the main memory in a computer.
- **DRAM** - Dynamic Random Access Memory : Are volatile memory that stores bits of data in transistors, this memory is located closer to your processor too, so your computer can easily and quickly access it for all the processes you do.
As you use your computer, it needs to recall data and programming code for the CPU to process. RAM provides a way for the computer to use rewrite and temporarily save the data and code in real time.
- **Memory management unit**: "MMU", ① is a hardware unit having all memory reference passed through itself.
② An MMU effectively performs virtual memory management, handling at the same time memory protection.



CPU

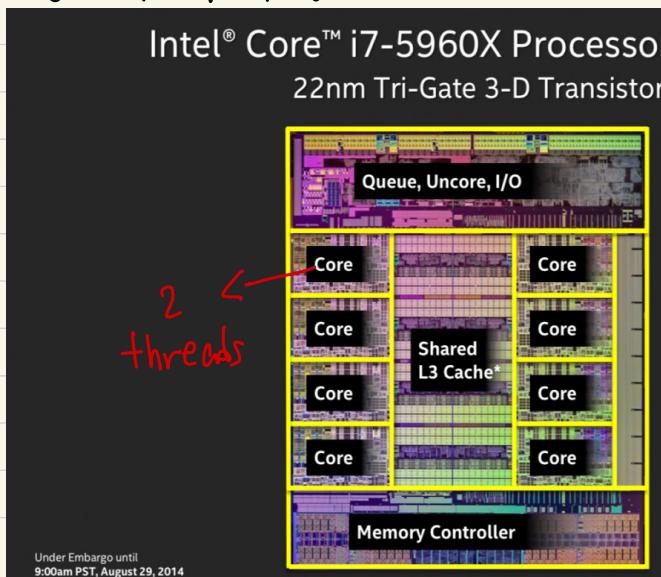
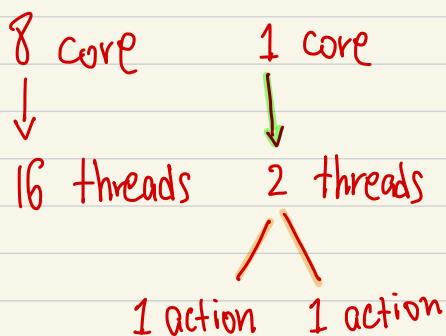
Central Processing Unit: it's the electronic circuitry the executes instructions comprising a computer program.

Multithreading and Multiprocessing

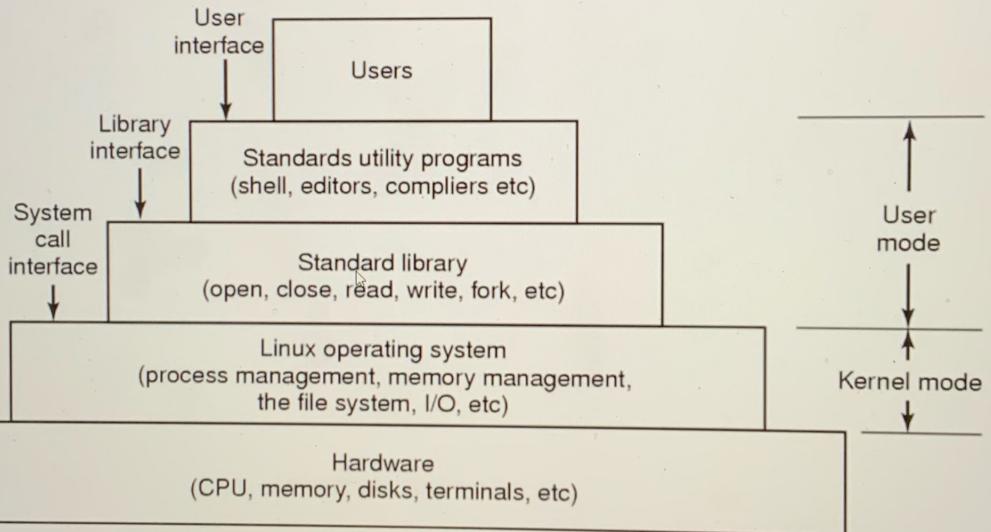
Multiprocessing: systems include multiple complete processing units in one or more cores.

Multithreading: aims to increase utilization of a single core by using thread-level parallelism, as well as instruction-level parallelism.

* As the two techniques are complementary, they are combined in nearly all modern systems architecture.



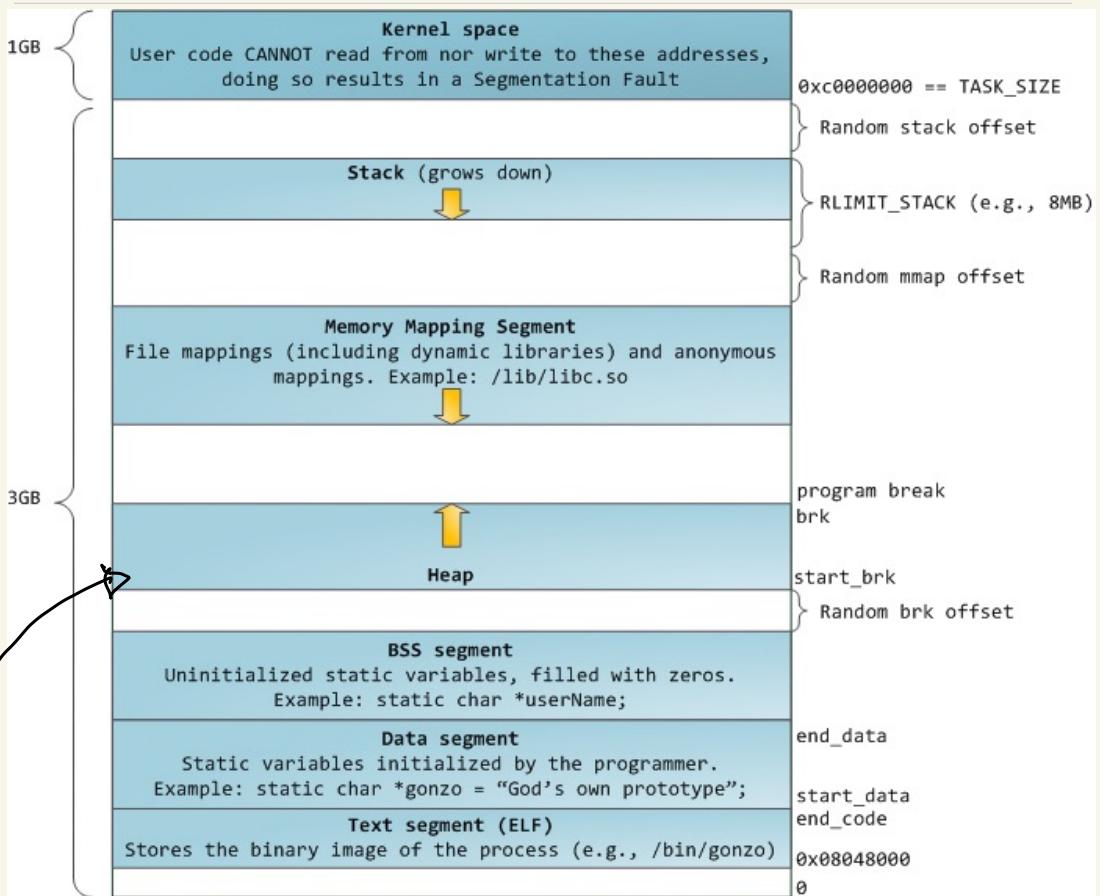
OS and programs



Process:

- instance of an executing program
- Each process has its own address space, which is protected from other processes by HW
- OS creates a sandbox for the process - makes it feel as if it "own" the whole computer.

Process memory map :



Heap: New class → Creates new object → the memory of this New object located on Heap

אנו נזכיר את הערך החדש שיצורנו בזיכרון החדש.

Process Creation

- UNIX - fork():
 - The child inherits all the environment variables, memory image (variables) and resources (e.g. open files) of the parent
 - The returned value is used to determine if it is a parent or child.
 - The child may use another system call to load and run a different executable
- Windows - CreateProcess():
 - Creates a new process, sets environment, loads and runs an executable in one go.

Process Termination

- Voluntary:
 - exit() system call in Unix
 - ExitProcess() in Windows
- Involuntary:
 - A process that makes a fatal error (e.g. division by zero) is signaled (interrupted) by the OS
 - Killed by another: kill() in UNIX and TerminateProcess() in Windows

kill: Send a signal to a process

Process Hierarchies:

Unix:

- Process has only one parent
 - Each child may have children
 - process cannot disinherit its child.

the systemd process

- The parent of all other processes is now the systemd process (pid=1)
 - The init process is rarely used these days.
 - The systemd should be thought as a complete init system, that is used to:
 - bootstrap user space and manage user processes
 - unify service configuration and behavior across Linux distributions

Systemd: Handles all startup of services

zombie process

- Process collects statistics about itself (PID, termination status, resource usage information)
- This information is made available to the parent process - using `wait()` - after the process dies
- If this information is not collected, the entry stays in the kernel space and is called a zombie.
- A zombie process is not running anymore!!!
It is dead, but not waited.

zombie process

- A zombie process created in python
- Run the interpreter and import the test file
- The created child process exits, but the parent will not `wait()` for it.

`test.py`

```
import os
import time

pid = os.fork()
if pid==0:
    time.sleep(3)
    exit()
```

```
1@osboxes:~$ ps -ux | grep 29889
1 29889 0.0 0.0      0 0 pts/1  Z+ 12:20  0:00 [python3] <defunct>
1 29936 0.0 0.0 17732 2228 pts/0 S+ 12:26  0:00 grep --color=auto 29889
1@osboxes:~$
```

Linux 1) I f d s o c u s l o c d e process

test.py

```
import os  
import time  
  
pid = os.fork()  
if pid==0:  
    time.sleep(100)  
    exit()  
else:  
    time.sleep(100)  
    exit()
```

orphan process

- An orphan process is a process (still running) whose parent process has terminated
- Orphan processes are “adopted” by some other system level process.
(process 1500 is part of systemd)

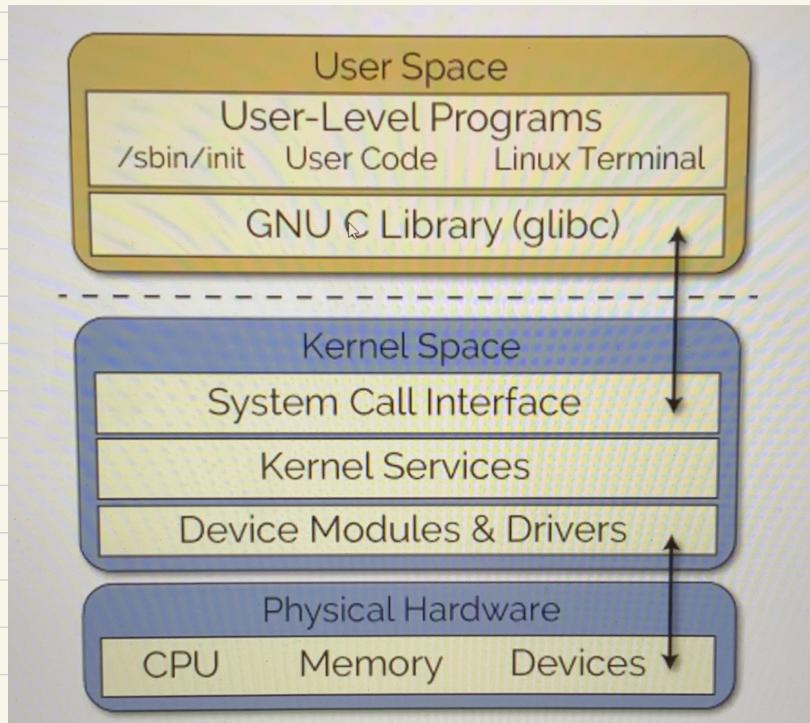


1	30138	29709	0	12:53	pts/1	00:00:00	python3
1	30139	30138	0	12:53	pts/1	00:00:00	python3

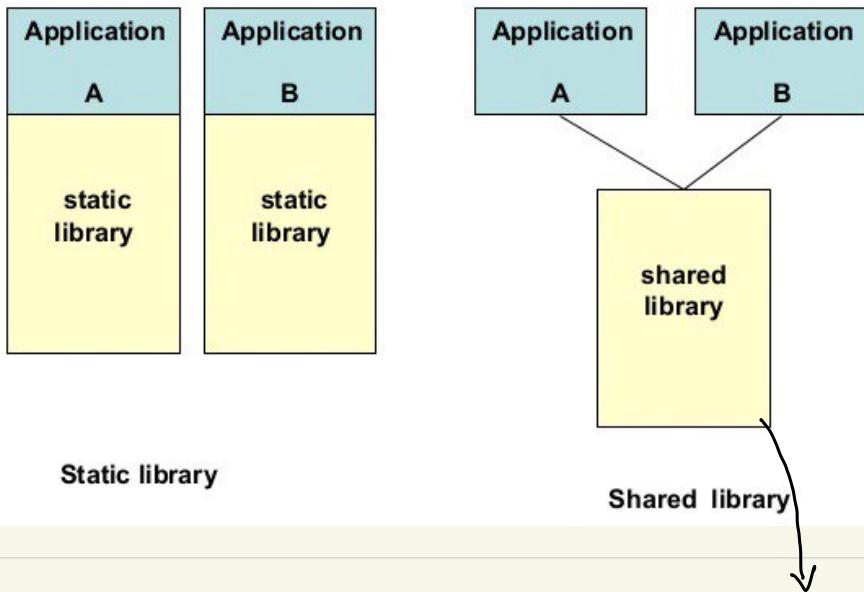
1	2131	1855	0	Sun Jun 10	:	00:00:11	/usr/bin/
1	30139	1500	0	12:53	pts/1	00:00:00	python3

System calls

- What is a system call?
 - API to the OS
 - Requests a service from an operating system's kernel
- There are 2 CPU modes:
 - User mode:
 - Limits the address space of the program.
 - Prevents the application from directly using devices
 - Kernel mode
 - All is allowed
- POSIX defines about 100 system calls:
open, read, write, close, wait, execve, fork, exit, kill
- All man 2 functions are system calls



Static Library vs. Shared Library



این جیز اینس، نیک نیک RAM -d یا بی کد

IT'S loaded into RAM once, and shared among the processes.