

Sequential vs. Multi Processing, Concurrent, Parallel and Distributed Computing



A Simple Machine

- A machine makes available computing and storage resources.
- By computing resource we mean, e.g., a processor, such as a CPU.
- By storage resource we mean, e.g., a given amount of primary memory.

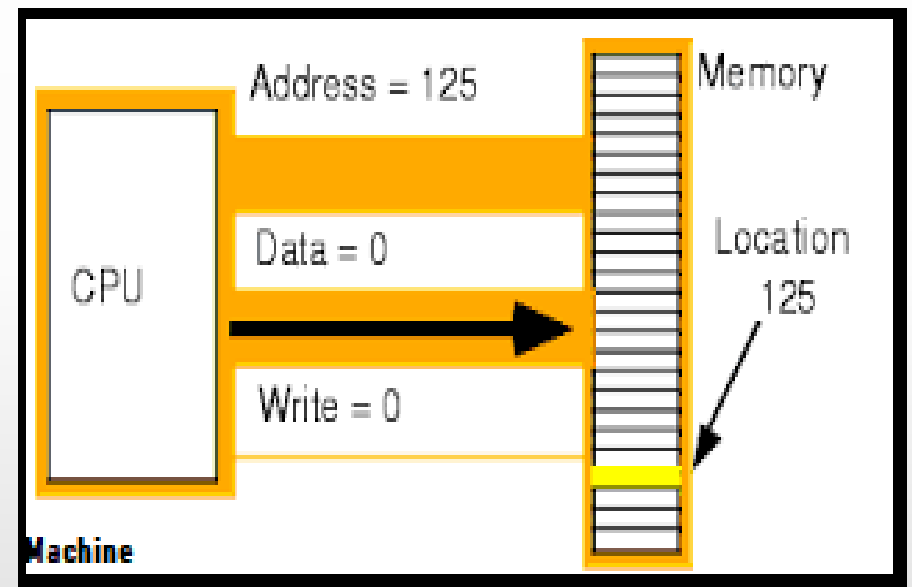
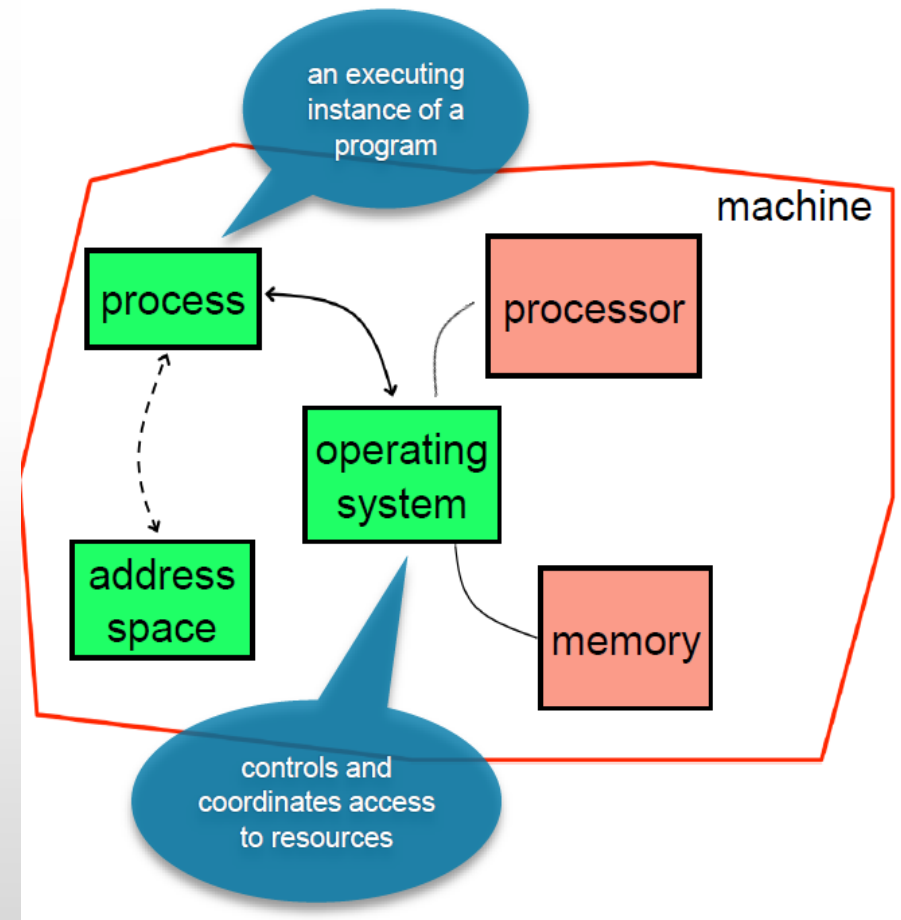


Figure Source: plantation-productions.com



- A process is an executing instance of a program.
- **Resource usage** is typically **controlled** by an operating system (**OS**).
- Since resources are scarce and differ in their capabilities, an OS aims to make **the most efficient use possible** of those resources.
- The OS assigns a **unique identity to each process** and then controls how a process is granted access to computing resources.
- The OS also controls how a process is granted access to storage resources by assigning an address space to that process.
- When the OS ensures that **each process P has a single address space A that is exclusive to P**, we are allowed a **sequential reading of the steps that comprise the process**.



Sequential reading of the steps that comprise the process

- The OS ensures that no other process tampers with x and y , and that only *foobar* has access to x and y .
 - Even if many applications are running, if we call *foobar*, we should expect the results of x and y to come out correctly, according to the input parameters given.

executes as a process

```
def foobar(a, b):
```

```
    x = a+1
```

```
    y = b+3
```

```
    x = y-x
```

```
    print 'x = ', x
```

```
    print 'y = ', y
```

x and y are locations in the address space allocated by the OS to foobar



Limitations

- What if processing to be done is more than just a simple sum?
- What if *foo* and *bar* take long to run?
- The processor may have to wait on slow peripheral usage in one or both of them, for example.
- Can we run them concurrently ?
- Perhaps one function belongs to Amazon (e.g., book data) and the other to Google (e.g., news)?

```
def foobar(a, b):  
    x = foo(a,1)  
    y = bar(b,3)  
    x = y-x  
    print 'x = ', x  
    print 'y = ', y
```



Sequential Processing

- Further, what if, instead of addition, the functions applied to a and b were
 - more computationally demanding than a single machine can cope well with?
 - not available locally in the machine where *foobar* is running?
- If *foo* and *bar* take long to run, we may wish to run them concurrently, in different processes, and perhaps even better, in parallel, in different processors.
 - If *foo* and *bar* are proprietary services held in remote machines, we may not be able to hold local copies of them.

Sequential, isolated processing is **simple** , but bounded and **limiting**.



- **Non sequential, non isolated** processing expands the bounds and limits with respect to **performance**.
- It is possible to switch between a process P and a process P' if, for example, P is idle waiting for something (like I/O) to complete; also, we get more responsiveness, as:
 - While printing a long document, your machine still allows you to go on doing other things.
 - While downloading a file, a web browser still allows you to traverse a link.
 - While you ponder what to do next, the same machine goes about attending to someone else.

