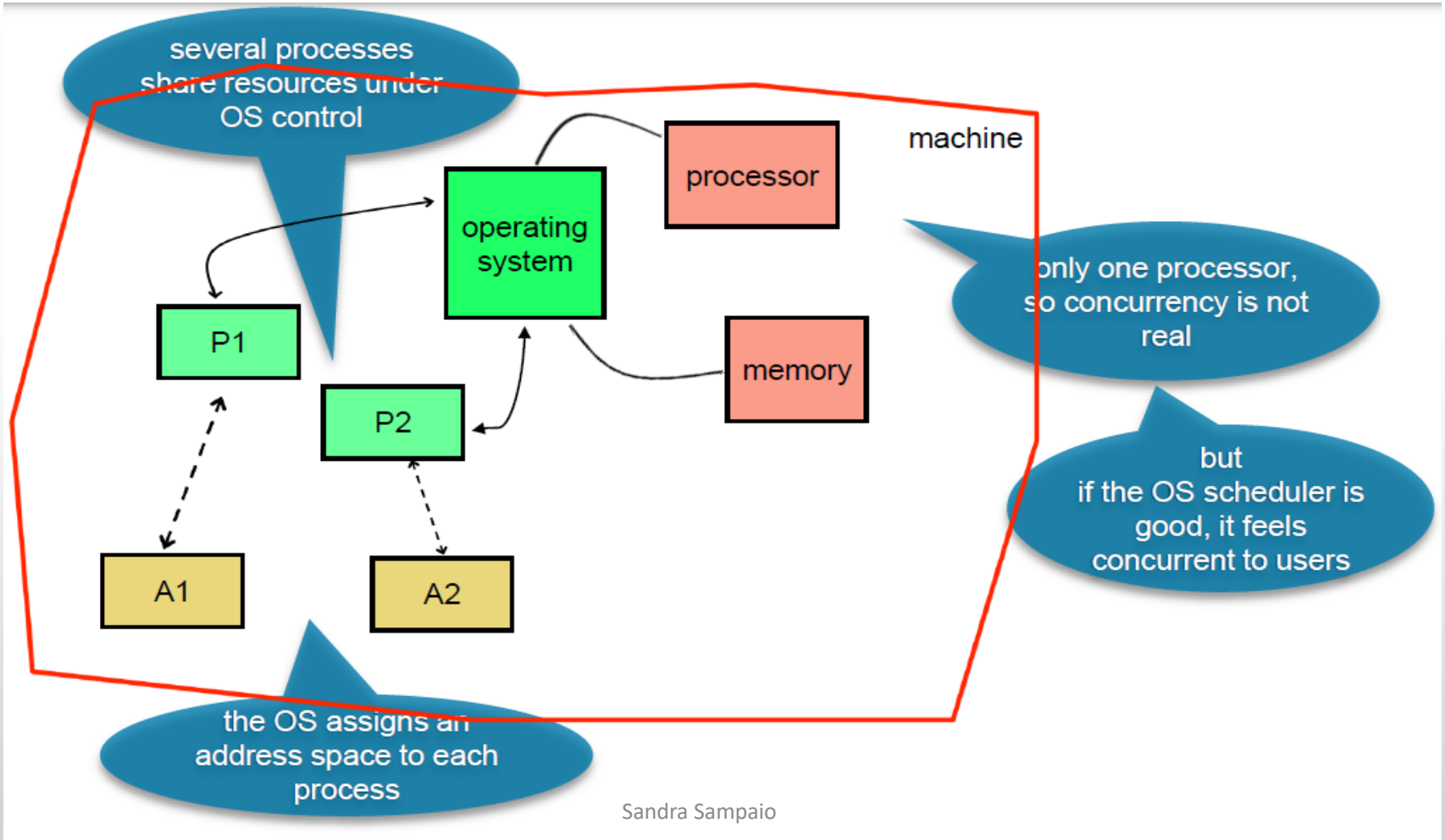
MANCHESTER 1824

The University of Manchester

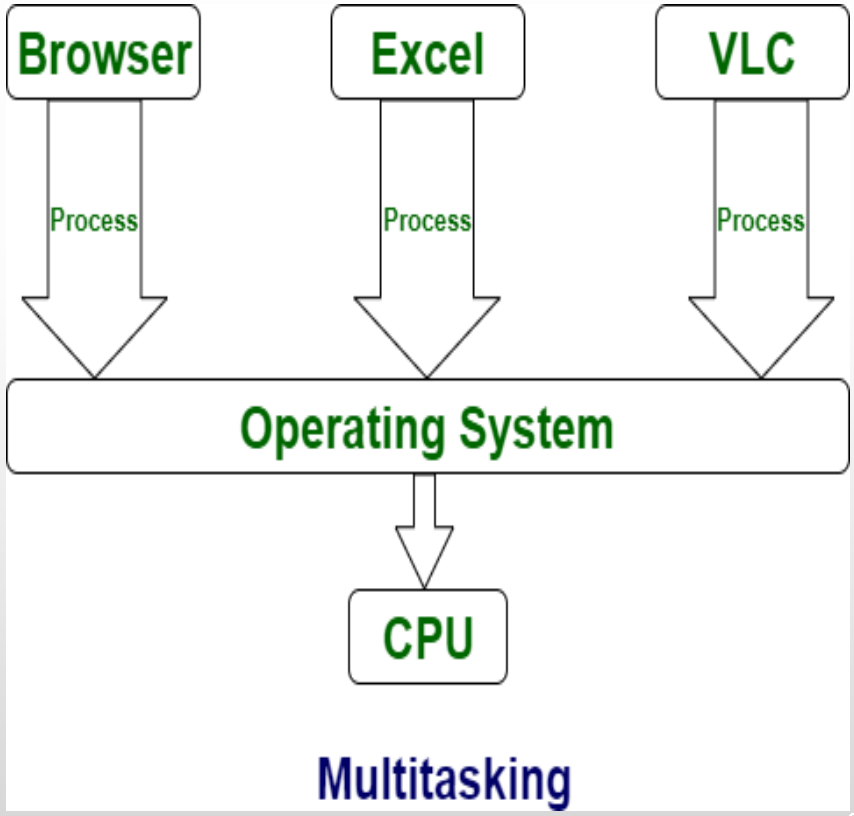
Sequential vs. Multi Processing, Concurrent, Parallel and Distributed Computing

Multi-Processing

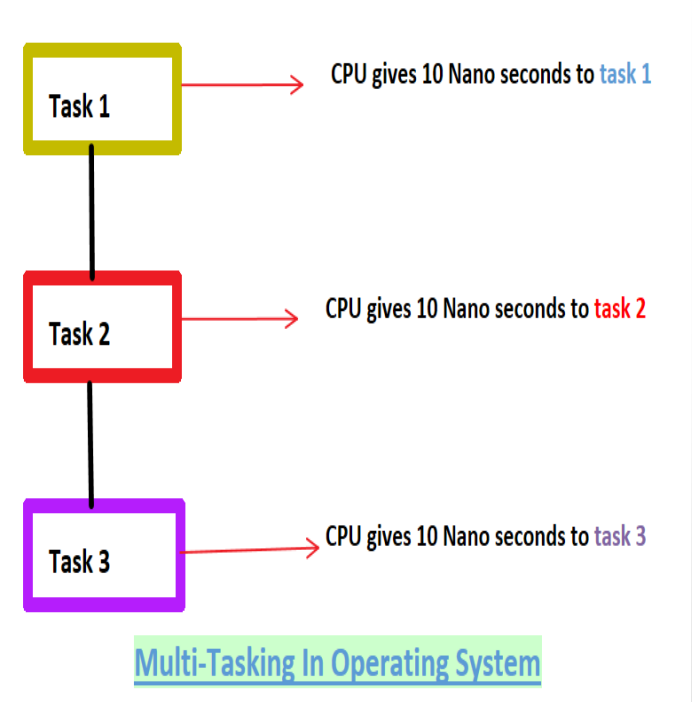


Multi-Processing vs. Multi-Tasking

- Two different concepts.
- An OS multi tasks by:
  - allowing more than one process to be underway by controlling how each one makes use of the resources allocated to it.
  - implementing a scheduling policy, which grants each active process a time slice during which it can access the resources allocated to it.

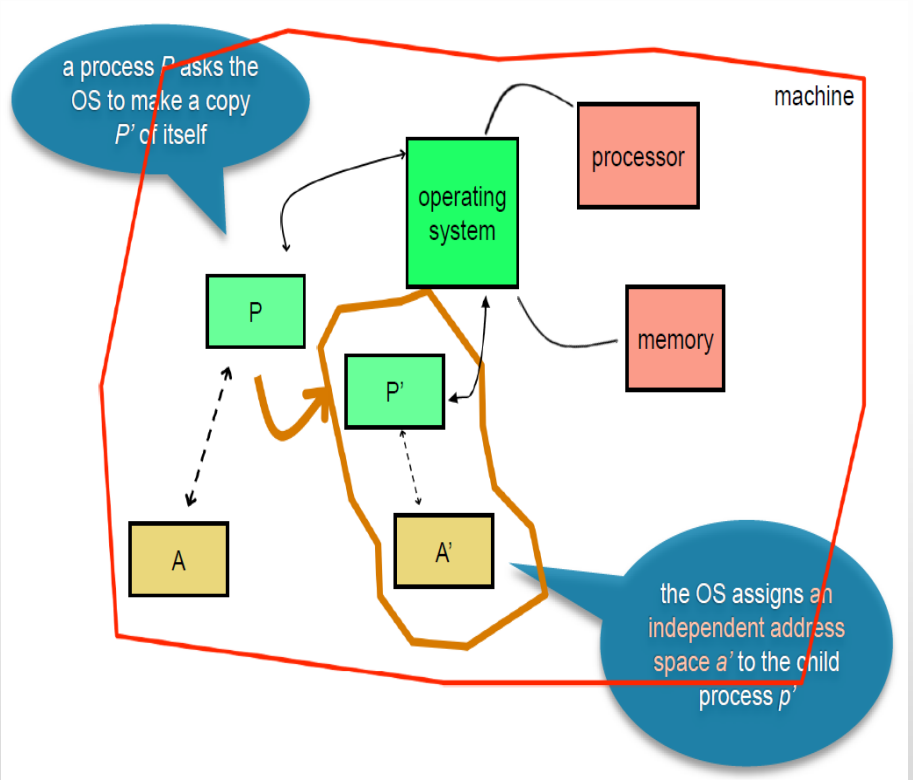


- Taking it literally, in multi-tasking, processes are not really executing concurrently.
  - Concurrent execution is only apparent.
- The appearance of concurrent execution stems from an effective scheduling policy.
- **If all processes get a fair share of the resource and they get it sufficiently often, it seems to users that all processes are executing concurrently.**
  - For example, while a process P is waiting on a slow output device, the OS may schedule another process P' to make use of the CPU. It seems to users that the machine is both printing for P and running P'.



Multi-Processing by Forking

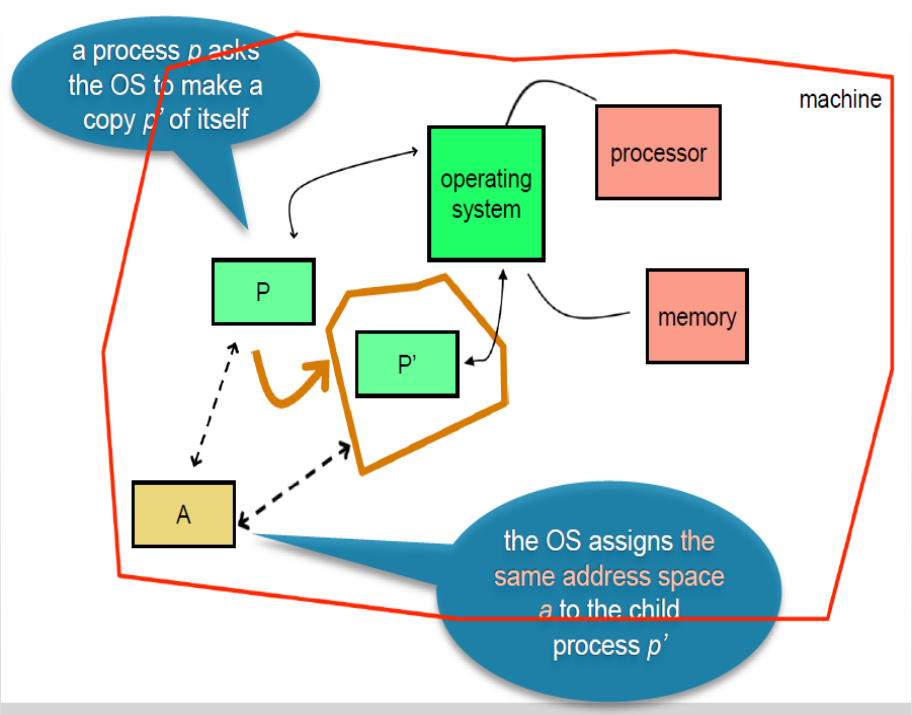
- When a process forks (using an OS call) it causes two copies of itself to be active concurrently.
- The child process is given a copy of the parent process's address space. The address spaces however are distinct. And so, if either process modifies a variable in its address space, this change is not visible to the other process.
  - The child process starts executing after the OS call.
  - The parent can continue or wait for the child to execute.
  - Ultimately the parent must mean to find out how and when the child completes execution.



- Forking is quite common in the client-server type of distributed computing, as a server typically forks a child process for each request it receives.
- Because of the copying , forking can be expensive.
- In practice, modern OSs have strategies that make the actual cost quite affordable.
- Forking is reasonably safe because the address spaces are distinct.
  - Discipline in adhering to best practice is nonetheless required (e.g., to avoid zombie processes, to avoid unintended sharing of references to files, etc.)

Multi-Processing by Threading

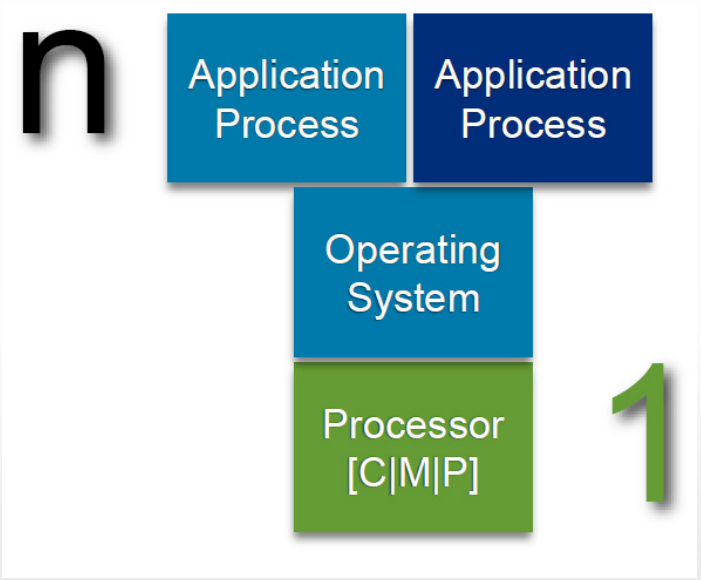
- Forking imposes a certain degree of isolation. And so, if parent and child need to interact and share, threading may be a better approach to multi tasking.
- With threading, the address space is not copied, it is shared.
  - This means that if one process changes a variable, all other processes see it.
  - This makes threading less expensive, but also less safe than forking.



Concurrent Computing

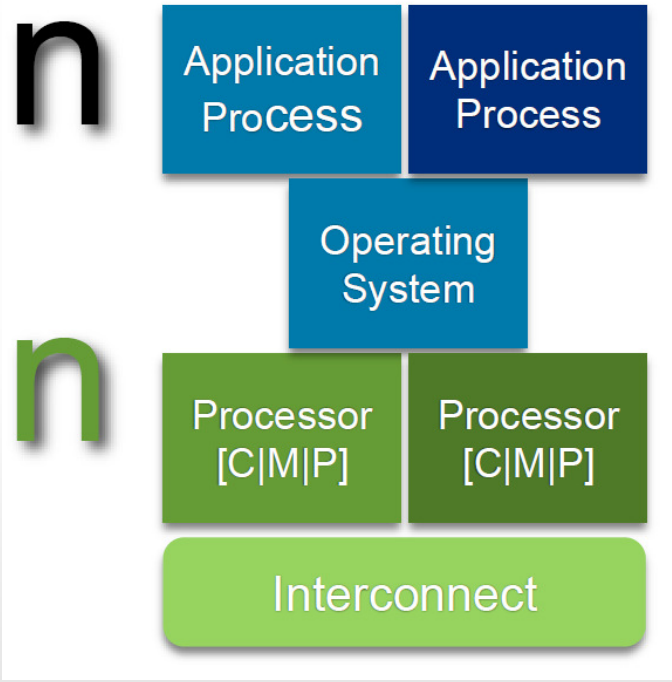
- Consider **many** application processes.
- Processes are often threads.

The OS schedules the execution of n copies of a process Pi, 1 =< i=< n, to run in the same processor, typically sharing a single address space.



Parallel Computing

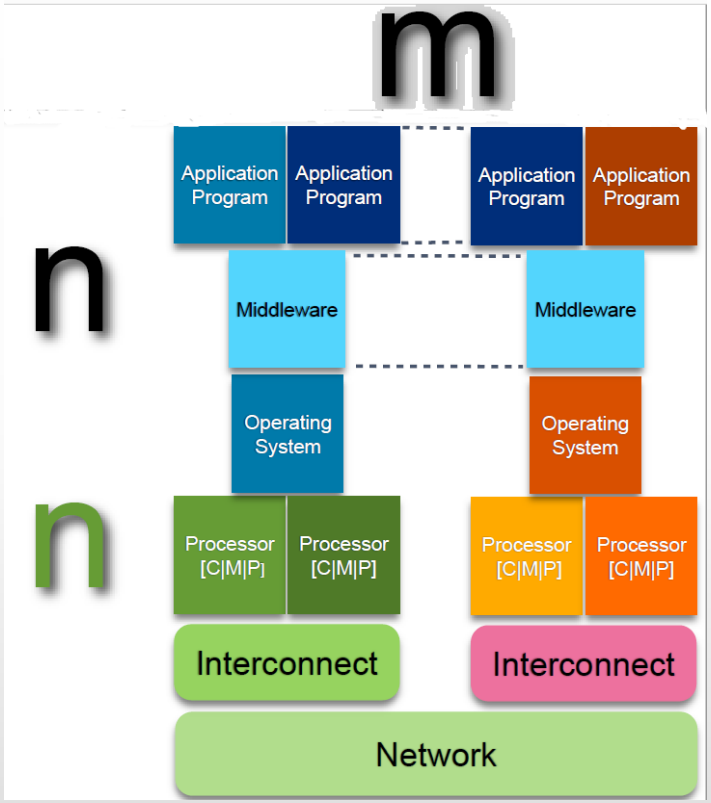
- There are now many processors bound by an interconnect (e.g., a bus across processors).
- There is **truly many** processes running at the same time, not just multi threading, but true parallelism.



The n copies of a process Pi, 1 =<i =< n, can, each, run in one of m, 1 =<j =< m, different processors Cj, possibly (but not necessarily) sharing a single address space A.

Distributed Computing

- There are many independent, self-sufficient, autonomous, heterogeneous machines.
- We now have spatial separation.
- Message exchange is needed, network effects are felt.
- Complexity may reach a point in which applications are not written against OS services. Instead, they are written against a middleware API. The middleware then takes some of the complexity upon itself.



The n (not necessarily identical) processes Pi, 1 =< i =< n, each run in one of m, 1 =< j =< m, different machines Mj, that cannot share a single address space A (and therefore must communicate).

Multi-Tasking vs. Multi-Threading vs. Multi-Processing

- Watch the following video to supplement your knowledge of these three concepts, which are often treated as synonyms, with a few examples using raspberry pies.

https://www.youtube.com/watch?v=Tn0u-IIBmtc