A Brief History

Sanzheng Qiao

Department of Computing and Software

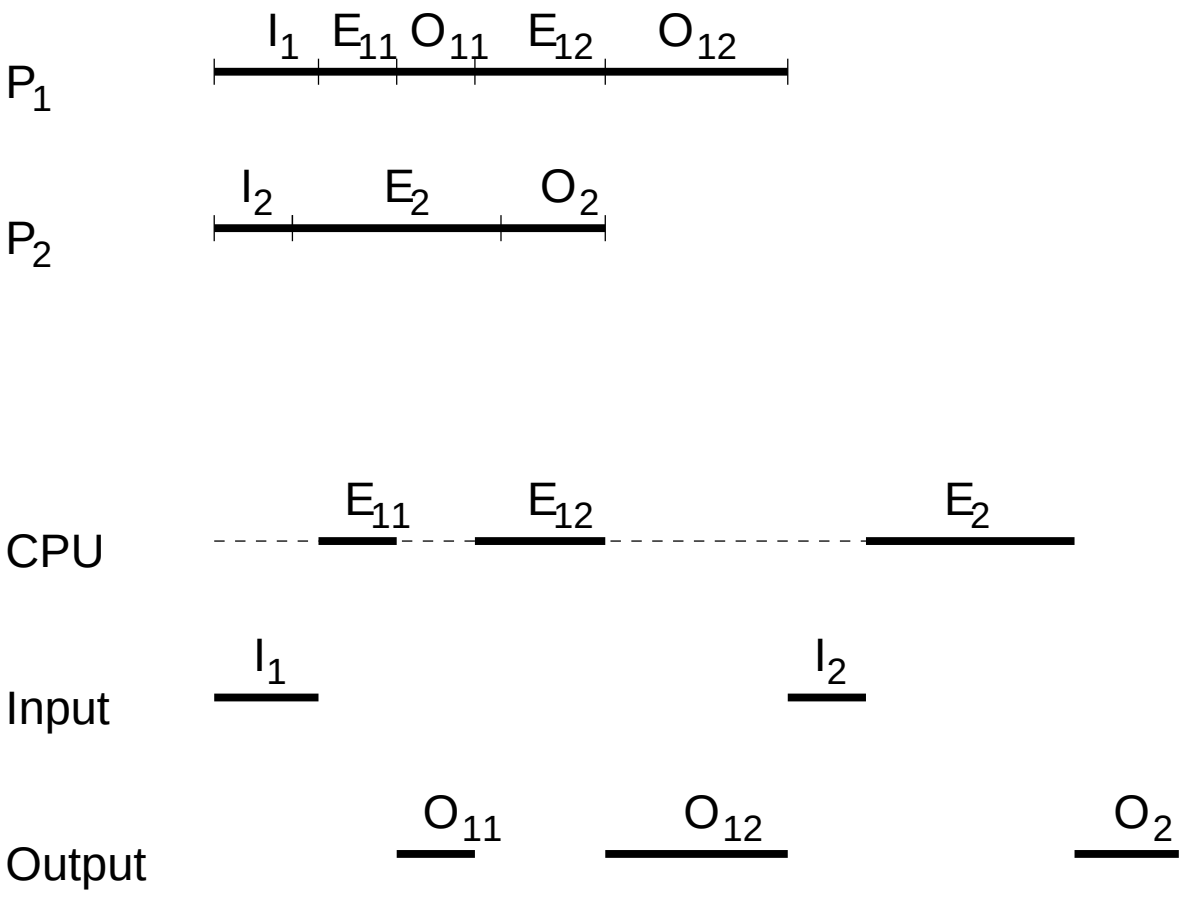
December, 2012

Batch processing

Systems that execute programs serially with no direct interaction between the user and the computer.

- The OS always resides in memory.
- Single user, no execution and I/O overlap.
- OS $=$ a program to load and run user jobs, take dumps,
- The CPU is often idle because I/O devices are much slower than the CPU.

Example

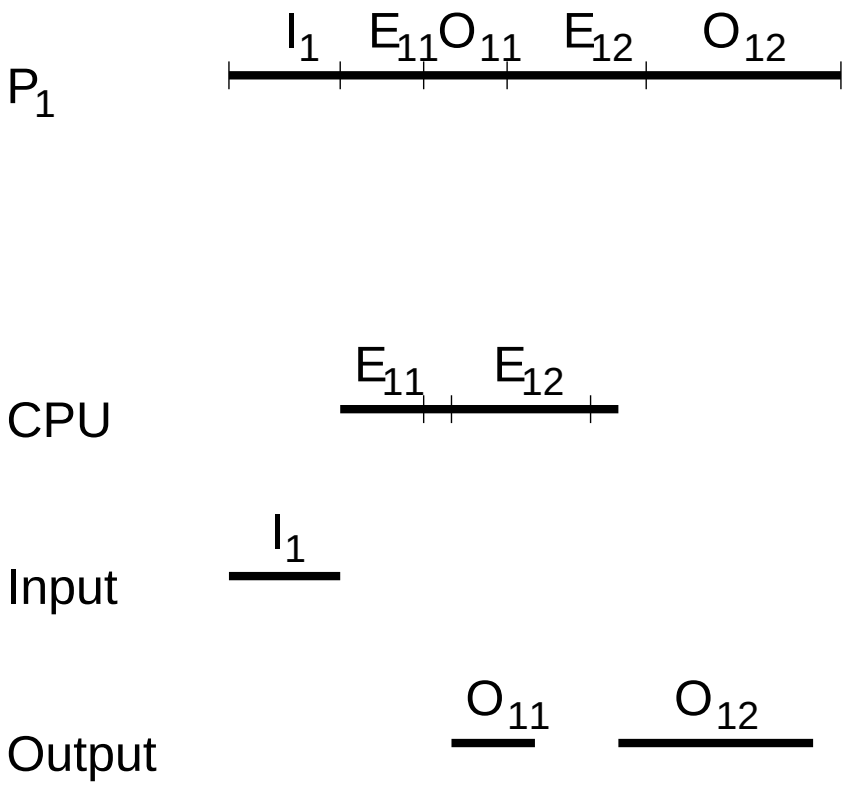

P1
I1
E11
O11
E12
O12
P2
I2
E2
O2
CPU
E11
E12
E2
Input
I1
I2
Output
O11
O12
O2

Buffering

Overlap the I/O of one job with its own computation.
I/O Interrupts The interrupt handler signals the user process upon the completion of I/O.

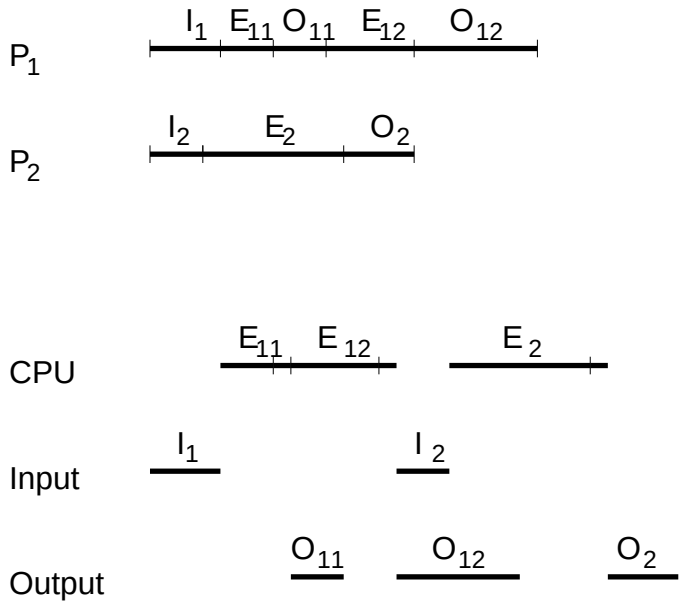
- Synchronous. The user process makes an I/O request and waits until the I/O completion. The user process must know the I/O latency, so it knows how long it should wait.
- Asynchronous. An I/O request returns without waiting for the I/O completion. An I/O interrupt is scheduled at the completion of I/O. The interrupt handler signals the user process when the I/O is done. This allows concurrent I/O operations to several devices.

Example



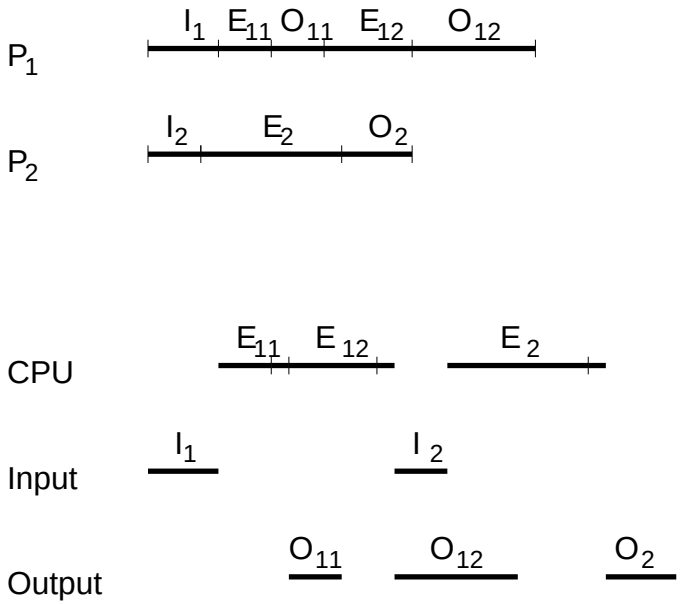
Multiprogramming

Systems that are designed to concurrently execute more than one task.



Multiprogramming

Systems that are designed to concurrently execute more than one task.

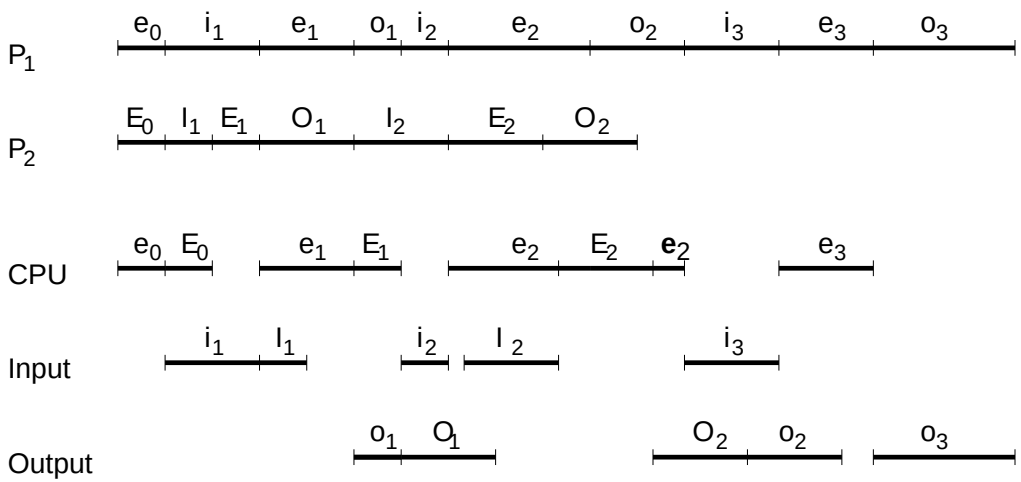


Issues:

- Memory protection + relocation.
- OS began to be an important science.

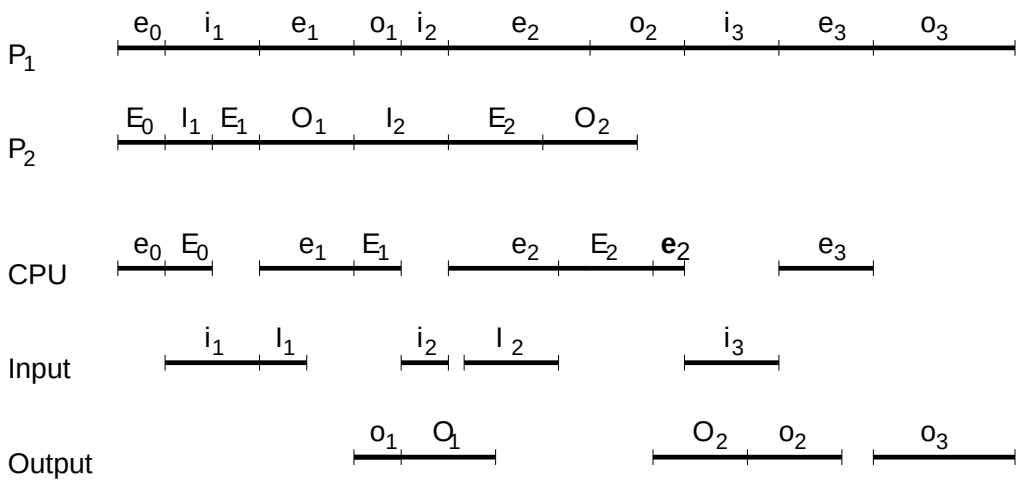
Timesharing or multitasking

Systems that allow multiple users (programs) to run concurrently. The system switches from one user to another. Examples, MULTICS at MIT and UNIX at Bell Lab (1970). Example.



Timesharing or multitasking

Systems that allow multiple users (programs) to run concurrently. The system switches from one user to another. Examples, MULTICS at MIT and UNIX at Bell Lab (1970). Example.



Issues:

- Response time.
- Thrashing.

Multiprocessor or multicore

High speed, better performance/price ratio, fault tolerant. Symmetric multiprocessing (SMP). All processors run the same OS, no master-slave relationship, share memory.

Multiprocessor or multicore

High speed, better performance/price ratio, fault tolerant. Symmetric multiprocessing (SMP). All processors run the same OS, no master-slave relationship, share memory.

Issues:

- data consistency.
- load balancing.
- I/O bottle-neck.

Cache coherency An update to a variable in one cache must be immediately reflected in all other caches that hold the variable.

Real-time

Operations have time limits on the operation of a processor or the flow of data. For example, sensor control.
Hard real-time system. Hard time limit on critical tasks.
Soft real-time system. A critical task get priority over other tasks and retains that priority until it completes.