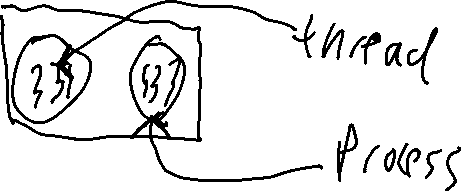
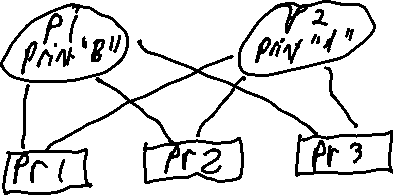
Into to Operating Systems – 17.10.19

* Lecture 5: Processor Management II: Concurrency Issues
* Processes are modular, and isolated from each other. This is to minimise interference between processes.
* A process has at least 1 thread, threads may be at user or kernel level. POSIX is the IEEE thread standard, threads share the resources of processes – files, memory





* Twin cores do not double the work, the speed is us not proportional as you would expect. – Amdahl’s Law
* The resource bottleneck: Multiple cores that share finite resources must be synchronised.
* **Introduction to Co-operating Processes**
* Processes within a system may be independent or co-operating.
* Independent processes cannot affect or be affected by the execution of another process.
* Co-operating processes can be affected or be affected by other processes, including sharing data.
* Issues with concurrent operation: Two processes accessing the same resources at the same time can lead to an over taxation of the resources, synchronisation is required due to this.
* 1. Incorrect Data: Doubled data, corrupted data, inconstant view of data.
* “Critical Region” – Designates an area that is locked away to a single process, until a configurable condition is completed. This prevents incorrect data.
* *“This thing can’t be accessed until this program is done with that area*



* “PA & PB want to print at the same time; both are referred to the IN (Shared memory) value?: *The person who comes last wins, as the last processes overwrites all others.*
* ^ If no synchronisation is done, this can result in lost files. Or programs acting unexpectedly.
* A Sync system will “Lock out” portions of memory to individual programs, allowing them to work in peace and not have their value overwritten by other programs – See “Critical Region”
* Mutual Exclusion – Program A must wait for program B to complete it’s process before
* Work in the “critical region” requires Atomic Operations
* 1: Buffer – Area of memory that stores data while it must wait for other processes to complete.
* **Read notes, she’s going too fast to note it down**
* “Deadlock” is when the processes are locked in by the scheduling system and cannot do anything. This leads to a total system standstill.
* Deadlock must be resolved via external intervention: I.E user
* Livelock is where processes are being moved around, however they are doing nothing.
* “Starvation” process or thread waits indefinitely. Deadlock and Livelock are forms of starvation.
* Deadlocks can occur on any type of filesystem, process system, etc