1 Threads

1.1 Theads from an OS perspective

A proces consists of two **fundamental units**

- **Resources**: all related resources are grouped together:
  - A logical addres space containing the process image (program, data, heap stack)
  - Files, I/O devices, I/O channels
- **Execution trace**, i.e, an entity that gets exceuted

A process can **share its resources** between **multiple execution traces**. Every thread has its own **execution context** (e.g program counter, stack, registers).
All threads have **access** to the process **shared resources**

- E.g files, one thread opens a file, all threads on the same process can acess the file
- Global variables, memory, etc. Which is needed or synchronisation

Some CPUs (hyperthreaded ones) have direct **hardware support** for **multi-threading**. They can offer up to 8 hardware threads per core.

Similar to processes, threads have:

- **States** and **transitions** (new, running, blocked, ready, terminated)
- **A thread control block**

Threads incur **less overhead** to create/terminate/switch (address space remains the same for threads of the same process)

Processes	Threads
Address space	Program Counter
Global variables	Registers
Open files	Stack
Child processes	State
Pending alarms	Local vars
Signals and signal handlers	
Accounting information	

- **Inter-thread communication** is easier/faster than **interprocess** communication (threads share memory by default)
- **No protection boundaries** are required in the address space. Threads are cooperating, belong to the same user, and have a common goal)
- **Synchronisation** has to be considered carfully

1.2 Why use threads

Multiple **related activites** apply to the **same resources**, these resources should be accessible/shared
Processes will often contain multiple **blocking tasks**

- I/O operations (thread blocks, **interrupt** marks completion)
- Memory access: pages faults are result in blocking

Such activities should be carried out in **parallel/concurrently**. **Application examples**: webservers, make program, spreadsheets, word processors, processing large data volumes