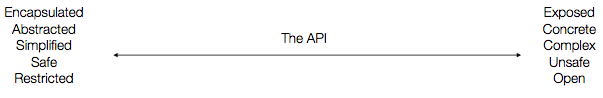
# ISU: 4. OS Api

**OS API – hvad er det?**

* Indkapsling af systemet
* 🡪 gjort tilgængelig med mere abstrakte kald
  + Simplifikation
* Platform uafhængigt



## The design philosophy - Why OO and OS Api?

* Abstraktionsniveau 🡪 jo højere abstraktionsniveau jo bedre (vil de fleste mene)
* F.eks.:
  + pthread\_create(&threadID, NULL, thread\_function, (void\*) args)
  + MyThread t1; t1.start();
* Platformsuafhængigt
  + Jo flere platforme vi kan programmere til jo bedre
  + 🡪 større portabilitet 🡪 hurtigere udvikling
  + **Portabilitet er keywordet**
  + Derudover kan man udvikle **til** en given platform, uafhængigt af hvilken platform der udvikles **på** 
    - Cross platform development
* **OS Api’et er blot et tyndt lag mellem OS og applikationen**
* Objekt orienteret fordi:
  + Nemmere at arbejde med - renere kode
  + Mindsker gap’et mellem design og implementering
    - The representational gap

## Elaborate on the challenge of building it and its current design

* Tilpasses til linux
  + Tråde: Posix thread
* Det er nødvendigt at interfacet til linux er nøjagtigt det samme som til Windows – ellers ryger idéen
  + Der kunne derfor designes ud fra windows interfacet
  + Samme kald med samme parametre 🡪 forskellen ligger i hvad der så sker i kaldet
* F.eks.:
  + I Mutex::lock, bliver pthread\_mutex\_lock(&mut\_); kaldt

## Effect on design/implementation

Ved brug af OS Api’et skal det ”fortælles” hvilket OS der skal compiles til. Ved at gøre dette inkluderes kun de nødvendige lib-filer til det givne OS.

* Det er nok bare at inkludere <OSApi/thread.hpp>
* Hvis OS\_LINUX er defineret, vil osapi/linux/thread.hpp blive inkluderet

### MQs (Message queues) used with pthreads contra MQ used in OO OS  Api.

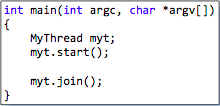
Nemmere at benytte message queues

### RAII in use

Resource acquisition is initilization

* I forbindelse med mutexes
* MessageQueues and messages
* Også RAII with unsubscribe!

### Using Threads before and now

* Væsentligt nemmere at benytte nu:
* 
* Tråd prioriteter!?
* Thread attributes