måndag 26 oktober 2020

Operativsystem

Föreläsning 1 - Introduktion

- Vad är ett operativsystem
  - Abstraktionsnivå, virualisering och resurshantering
  - Abstraktionsnivå
    - Hur skapar vi en abstraktionsnivå där vi kan programmera en process?
  - Virtualisering
  - Resurshantering
    - Tilldelar resurser på ett effektivt sätt, framförallt nödvändigt när flera olika resurser används samtidigt. Hur tilldelas dem mest effektivt mellan dessa program
- Abstraktion
  - När vi skriver ett program så gömmer operativsystemet komplexiteten och vi abstraherar komplexiteten med ett gränssnitt
  - Gränssnittet ger oss en miljö som är enklare att hantera
  - Vi kan då skriva program som arbetar mot operativsystemet, oberoende av hårdvaran, operativsystemet arbetar sedan med hårdvaran.
  - Operativsystemet kan dock hindra oss från att arbeta med hårdvaran på ett optimalt sätt
  - Vi abstraherar även från exakt vilken hårdvara det är till en instruktion set architecture
    - Ex: x86_64
  - Kernel i opsystemet arbetar mot hårdvaran
  - Vårt program arbetar mot våra biliotek
  - Operativsystemet utgörs av kernel och libraries eller bibliotek
  - Det operativsystem vi kommer att fokusera på är POSIX (Linux, Unix…)
  - MacOS är POSIX-kompatibelt
  - Vad består POSIX av:
    - Operating system API

    - Process handling: fork, exec, wait
    - Process communication: pipes
    - Threads handling: pthreads_create
    - Managing directory
    - Network handling
    - ….
  - The C standard library
    - Memory allocation: malloc, free
    - Signal handling
    - File operations
    - …
  - Command line interpreter
    - Shell: text based interface
    - Scripting languages (kommer ej kolla på i kursen)
  - Dessa tre ben utgör vår abstraktionsnivå
- Virtualisering
  - 2 processer som ligger och kör, båda tror att man kör på den faktiska hårdvaran men dem arbetar i en virtuell värld
  - Kan använda hårddisken och ramminnet för att skapa en illusion för programmet att den har mer RAM-minne att arbeta med än vad den har
  - Hypervisor: Kan alternera mellan olika operativsystem som är oberoende av varandra
  - Resursmanagement:
    - Time
      - Scheduling, dividing execution time among processes
    - Memory
      - Efficient allocation/deallocation, malloc/free