

## 2. Memory Protection

### Hvad er memory protection?

---

- Benyttes til at kontrollere rettigheder til forskellige dele af memory
- Hovedformålet er at forhindre processer i at få adgang til dele af memory den ikke har rettigheder til at tilgå
- Sørger blandt andet også for segmentation faults

### Hvordan fungerer en MMU?

---

- Typisk indbygget i CPU'en
- Sørger for at et program afvikles i sin egen private memory space
- Gør virtuel memory mulig, som er uafhængig af fysisk memory
- MMU'en er en oversætter fra virtuelle adresser til fysiske adresser
- Dette muliggør at programmer afvikles med de samme virtuelle adresser, mens de faktisk er på forskellige fysiske adresser
- De virtuelle adresser tildeles af compiler og linker, MMU'en sørger for at allokere dem i den fysiske hukommelse
- En af grundstenene for simple/organiseret multitasking
- **Translation Lookaside Buffer(TLB)**
  - Cacher nyeligt tilgængelige pages
  - Benytter main memory til at gemme information om virtuelle memory maps(PTE - Page Table Entry)
  - Indeholder information om fysisk adresse, størrelse, rettigheder, cache

### Hvordan laver man context switching vha en MMU?

---

1. Gem den aktive process' indhold og sæt den i dormant state(sleep)
2. Ryd cache
3. Ryd TLB
4. Opdater med den nye process' page tables
5. Genskab den nye process

### Hvordan benytter Linux sig af en MMU?

---

- Linux kan ikke afvikles uden en MMU
- MMU'en sørger for opdelingen af kernel-/user-space
- Virtuel memory er en hoved del af Linux

- Segmentation faults
- Data beskyttelse

