

Objekt-Orienteret Programmering

Basics

Aslak Johansen asjo@mmmi.sdu.dk
Peter Nellesmann pmn@mmmi.sdu.dk

September 4, 2024

Part 0: Grafer

Grafer ▷ Knuder og Kanter

En graf er en hyppigt anvendt form for datastruktur.

Grafer ▷ Knuder og Kanter

En graf er en hyppigt anvendt form for datastruktur.

Den mest basale graf består af:

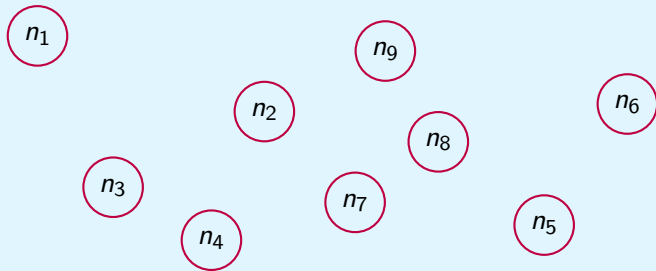
- ▶ **Knuder** (eng: **nodes** | **vertices**) Ofte tegnet som en cirkel eller kasse.
- ▶ **Kanter** (eng: **edges**) Illustreret som en linje eller pil mellem to knuder.

Grafer ▷ Knuder og Kanter

En graf er en hyppigt anvendt form for datastruktur.

Den mest basale graf består af:

- ▶ **Knuder** (eng: **nodes** | **vertices**) Ofte tegnet som en cirkel eller kasse.
- ▶ **Kanter** (eng: **edges**) Illustreret som en linje eller pil mellem to knuder.

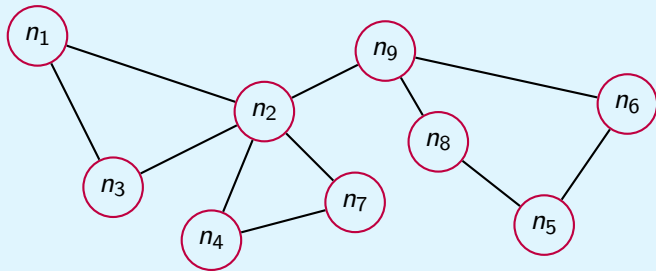


Grafer ▷ Knuder og Kanter

En graf er en hyppigt anvendt form for datastruktur.

Den mest basale graf består af:

- ▶ **Knuder** (eng: **nodes** | **vertices**) Ofte tegnet som en cirkel eller kasse.
- ▶ **Kanter** (eng: **edges**) Illustreret som en linje eller pil mellem to knuder.

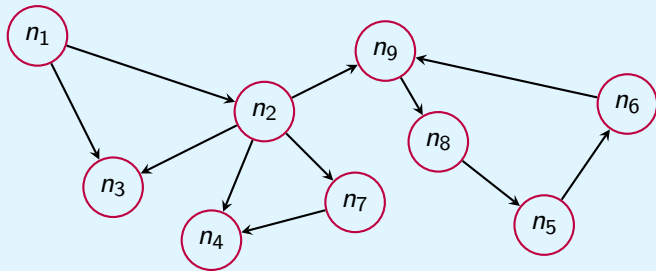


Grafer ▷ Knuder og Kanter

En graf er en hyppigt anvendt form for datastruktur.

Den mest basale graf består af:

- ▶ **Knuder** (eng: **nodes** | **vertices**) Ofte tegnet som en cirkel eller kasse.
- ▶ **Kanter** (eng: **edges**) Illustreret som en linje eller pil mellem to knuder.



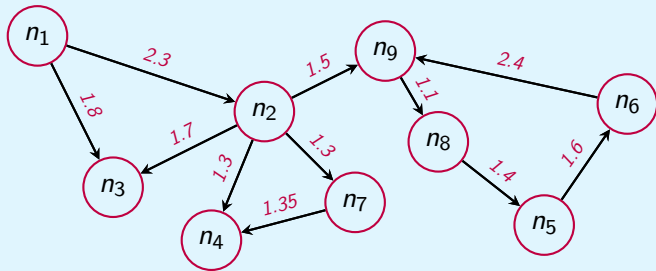
Kanter kan have en retning. Kanten har dermed en kildeknude (eng: **source node**) og en destinationsnode (eng: **destination node**). Vi kalder dermed grafen for *orienteret* (eng: *directed*).

Grafer ▷ Knuder og Kanter

En graf er en hyppigt anvendt form for datastruktur.

Den mest basale graf består af:

- **Knuder** (eng: **nodes** | **vertices**) Ofte tegnet som en cirkel eller kasse.
- **Kanter** (eng: **edges**) Illustreret som en linje eller pil mellem to knuder.



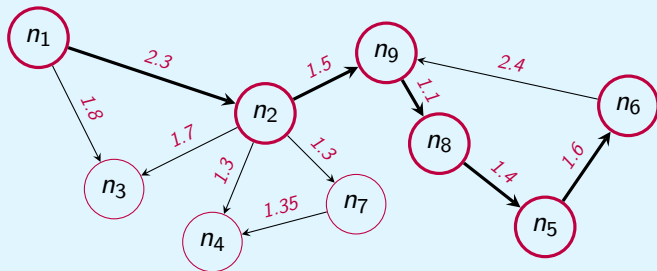
Kanter kan være associeret med en værdi (eng: **weight**). Hvis alle kanter er associeret med en værdi kalder vi den for **weighted**.

Grafer ► Stier

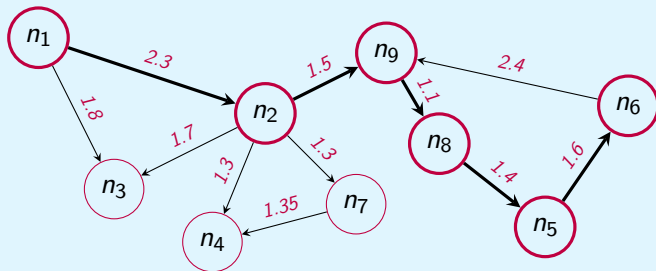
En *sti* (eng: **path**) er en serie af kanter der er forbundet igennem knuder. Alle kanter i denne sekvens skal pege i sekvensens retning (hvis orienteret).

Grafer ▷ Stier

En *sti* (eng: **path**) er en serie af kanter der er forbundet igennem knuder. Alle kanter i denne sekvens skal pege i sekvensens retning (hvis orienteret).



En *sti* (eng: **path**) er en serie af kanter der er forbundet igennem knuder. Alle kanter i denne sekvens skal pege i sekvensens retning (hvis orienteret).



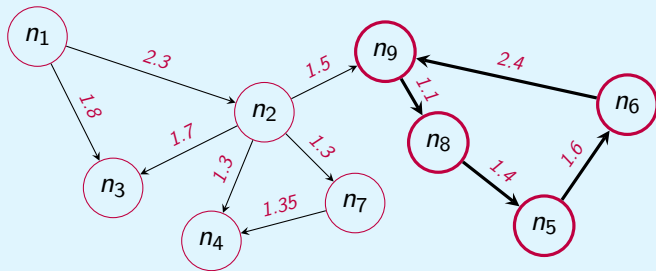
Dette kan eksempelvis bruges til at udregne distancen mellem n_1 og n_6 som $2.3 + 1.5 + 1.1 + 1.4 + 1.6 = 7.9$.

Grafer ▷ Kredse

En *kreds* (eng: **cycle**) er en sti hvor startknuden er den samme som slutknuden.

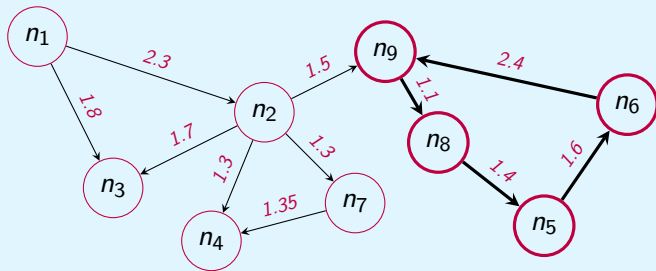
Grafer ► Kredse

En *kreds* (eng: **cycle**) er en sti hvor startknuden er den samme som slutknuden.



Grafer ► Kredse

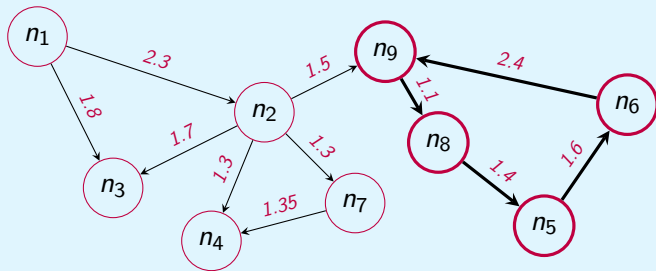
En *kreds* (eng: **cycle**) er en sti hvor startknuden er den samme som slutknuden.



Da det er besværligt at arbejde med grafer der har kredse er dette ofte en egenskab som man ønsker at undgå.

Grafer ▷ Kredse

En *kreds* (eng: **cycle**) er en sti hvor startknuden er den samme som slutknuden.



Da det er besværligt at arbejde med grafer der har kredse er dette ofte en egenskab som man ønsker at undgå.

En meget populær klasse af grafer kaldes for *directed acyclic graphs* (eller kort: **DAGs**).

Part 1:

Træstrukturer

Træstrukturer ▷ Definition

En træstruktur er en minimalt udspændende, svagt forbunden, orienteret graf hvor alle knuder kan nås ud fra én rodknude ved at følge en sti af udgående kanter.

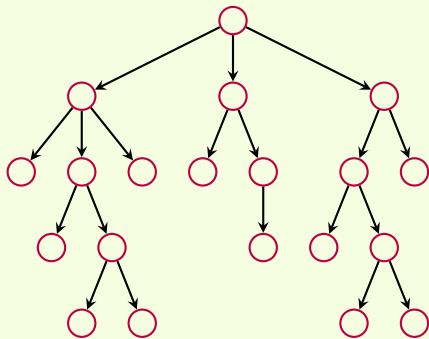
Træstrukturer ▷ Definition

En træstruktur er en minimalt udspændende, svagt forbunden, orienteret graf hvor alle knuder kan nås ud fra én rodknude ved at følge en sti af udgående kanter.

Dette betyder:

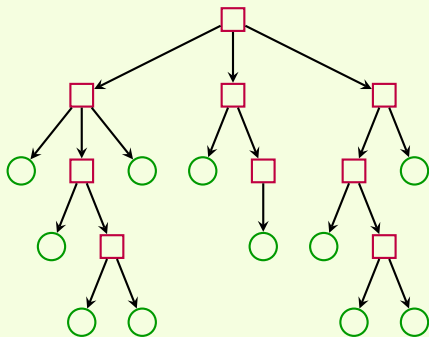
- ▶ **Graf** Se tidligere slides.
- ▶ **Orienteret** Kanterne har en retning.
- ▶ **Svagt forbunden (eng: weakly connected)** Der eksisterer en sti (hvis man undlader at betragte retningen) fra en vilkårlig knude til en vilkårlig anden knude.
- ▶ **Rodknude (eng: root node)** En knude der ikke har nogen kanter der peger på den.
- ▶ **Udgående kanter** De kanter der har den aktuelle knude som kilde.
- ▶ **Minimalt udspændende** Man kan ikke fjerne en kant uden at bryde én af ovenstående.

Træstrukturer ▷ Knude Typer



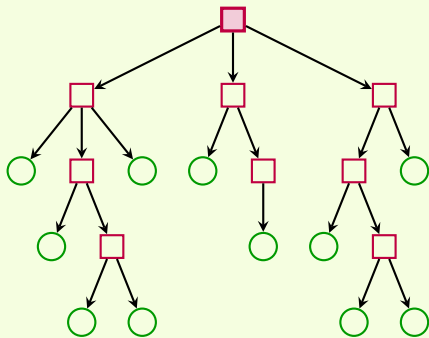
Eksempel på en træstruktur. Træstrukturer tegnes typisk *på hovedet*.

Træstrukturer ▷ Knude Typer



Nogle knuder er forgreninger (eng: **branch nodes**), andre er blade (eng: **leaf nodes**).

Træstrukturer ▷ Knude Typer



Én knude udgør roden (eng: **root node**).

Part 2: Filsystemer

Filsystemer ► Rolle

Vores styresystem bruger filsystemer til at organisere, og holde styr på, filer.

Filsystemer ▷ Rolle

Vores styresystem bruger filsystemer til at organisere, og holde styr på, filer.

De fleste filsystemer er organiseret omkring en træstruktur:

- ▶ ***Forgreninger*** Er vores foldere (teknisk eng: **directories**).
- ▶ ***Blade*** Er vores filer.

Filsystemer ► Rolle

Vores styresystem bruger filsystemer til at organisere, og holde styr på, filer.

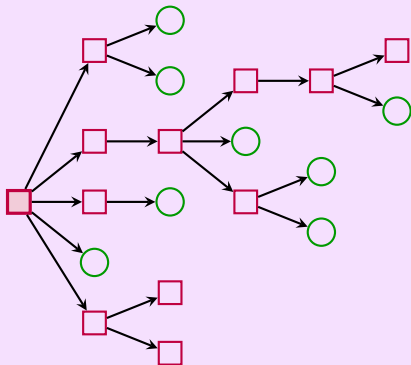
De fleste filsystemer er organiseret omkring en træstruktur:

- **Forgreninger** Er vores foldere (teknisk eng: **directories**).
- **Blade** Er vores filer.

Bemærk: I praksis er det lidt mere kompliceret:

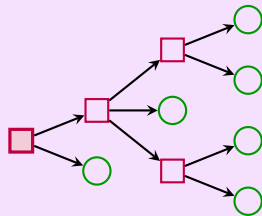
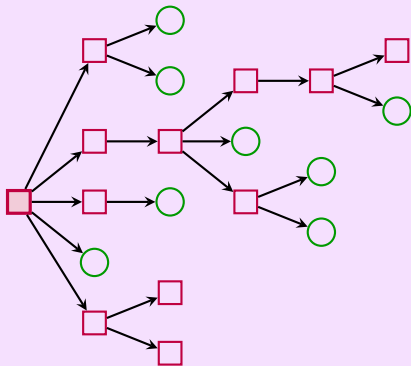
- Filer har metadata (herunder ejerskab og rettigheder).
- Moderne filsystemer har lænker (eng: **links**) og tillader at man *monterer* (eng: **mount**) andre filsystemer i dem.
- På UNIX-baserede systemer repræsenteres perifere enheder som en særlig type filer.
- På DOS-baserede systemer (e.g., Windows) kaldes første lag for “drev” (eng: **drives**).

Filsystemer ▷ Eksempel



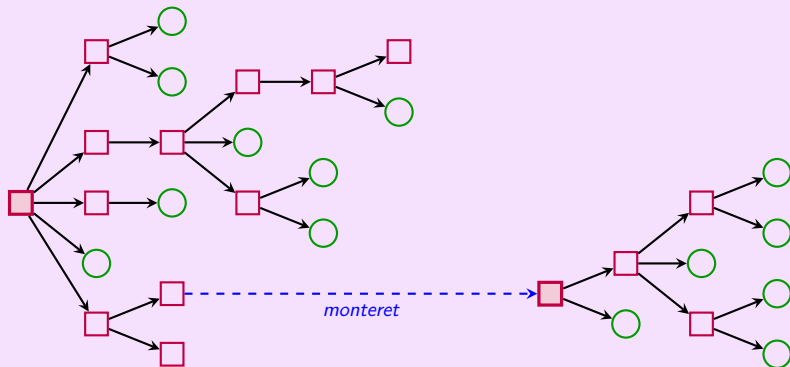
Eksempel på et rod-filsystem.

Filsystemer ▷ Eksempel



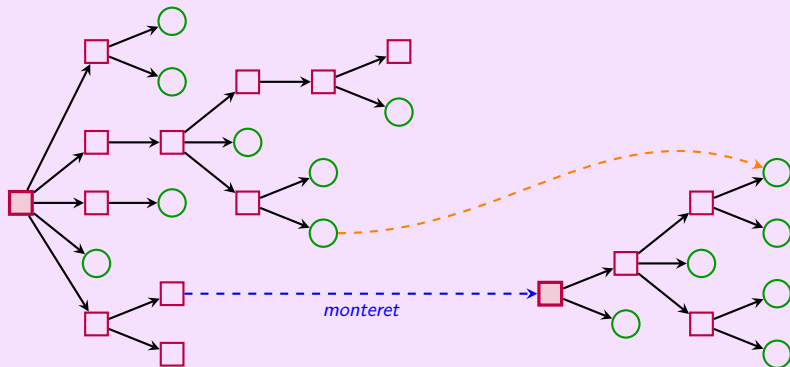
Et andet filsystem, fx på en USB stick.

Filsystemer ▷ Eksempel



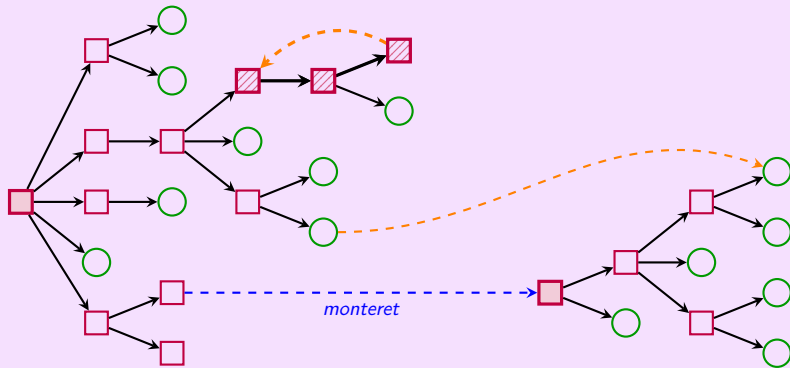
Det andet filsystem kan blive monteret (eng: **mounted**) i en folder (teknisk eng: **directory**).

Filsystemer ▷ Eksempel



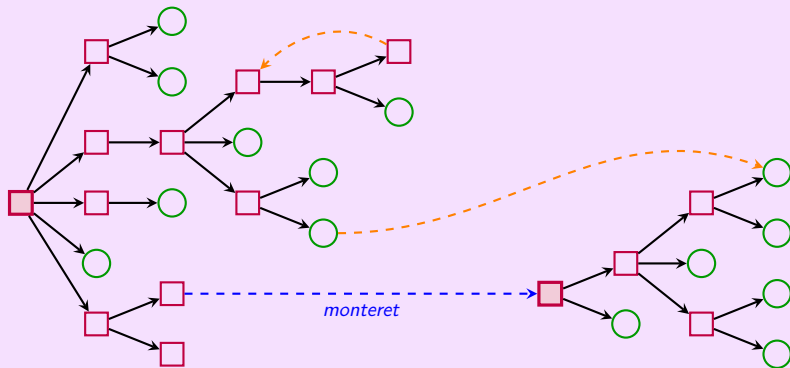
Nogle filsystemer understøtter *links* (hvis brug gør at filesystemet ikke længere er en træstruktur).

Filsystemer ▷ Eksempel

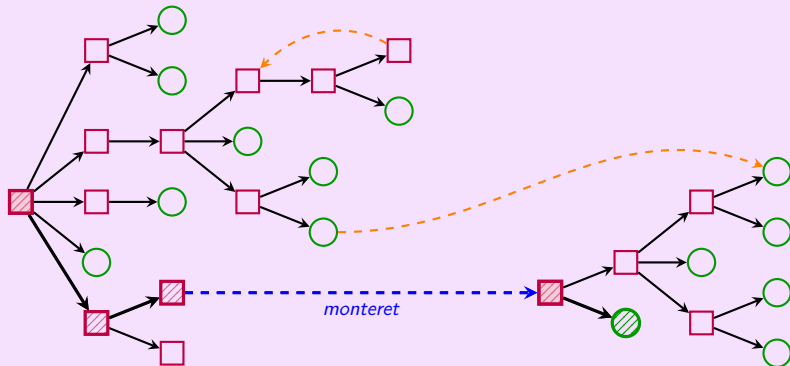


Med links kan man lave kredse i filsystemet.

Filsystemer ▷ Eksempel

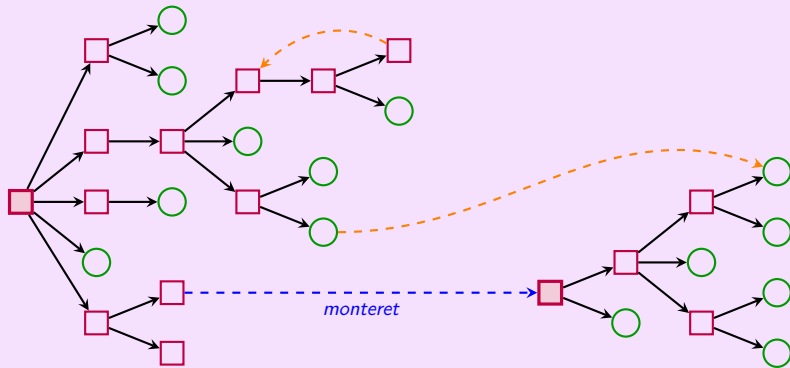


Filsystemer ▷ Eksempel



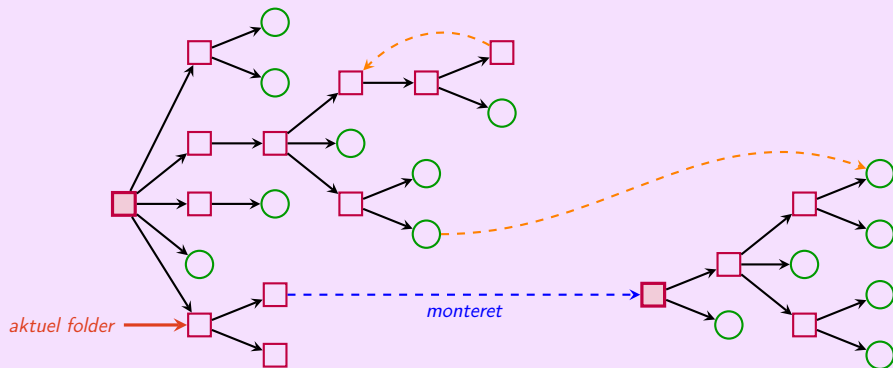
En position i filsystemet kan udpeges igennem en *absolut sti*, med udgangspunkt i træets rodnode.

Filsystemer ▷ Eksempel



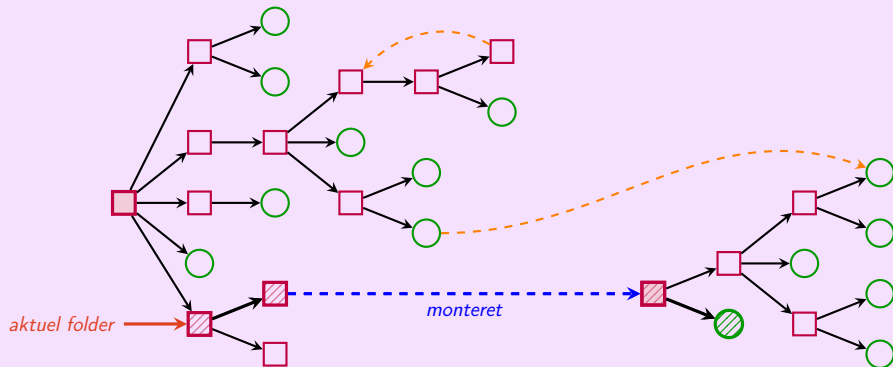
En position i filsystemet kan udpeges igennem en *relativ sti*, med udgangspunkt i “den aktuelle folder” (teknisk engelsk: **working directory**).

Filsystemer ▷ Eksempel



En position i filsystemet kan udpeges igennem en *relativ sti*, med udgangspunkt i “den aktuelle folder” (teknisk engelsk: **working directory**).

Filsystemer ▷ Eksempel



En position i filsystemet kan udpeges igennem en *relativ sti*, med udgangspunkt i “den aktuelle folder” (teknisk engelsk: **working directory**).

Filsystemer ► Filer

En fil ligger på et bestemt sted i et filsystem.

Filsystemer ► Filer

En fil ligger på et bestemt sted i et filsystem.

Den kan navngives ved en absolutte eller relative stier.

Filsystemer ► Filer

En fil ligger på et bestemt sted i et filsystem.

Den kan navngives ved en absolutte eller relative stier.

Samlet kalder man disse for *kvalificerede stier*.

Filsystemer ▷ Filer

En fil ligger på et bestemt sted i et filsystem.

Den kan navngives ved en absolutte eller relative stier.

Samlet kalder man disse for *kvalificerede stier*.

Filer er typisk associeret med metadata:

- ▶ Ejerskab.
- ▶ Rettigheder.
- ▶ Tidsstempler, eksempelvis for oprettelse, modifikation og tilgang.

Filsystemer ▷ Filer

En fil ligger på et bestemt sted i et filsystem.

Den kan navngives ved en absolutte eller relative stier.

Samlet kalder man disse for *kvalificerede stier*.

Filer er typisk associeret med metadata:

- ▶ Ejerskab.
- ▶ Rettigheder.
- ▶ Tidsstempler, eksempelvis for oprettelse, modifikation og tilgang.

Filens indhold er en sekvens af bytes. Vi kan læse dem én af gangen, eller indeksere ind i sekvensen. Hver byte kan (fx) tolkes som et tegn (uden formatering).

Part 3: Programmer

Programmer ▷ Kopiering af Fil

Programmer ▷ Kopiering af Fil

Kilde
Fil



Programmer ▷ Kopiering af Fil

Kilde
Fil



Destinations-
Fil



Programmer ▷ Kopiering af Fil

Kilde
Fil



Program
Hukommelse



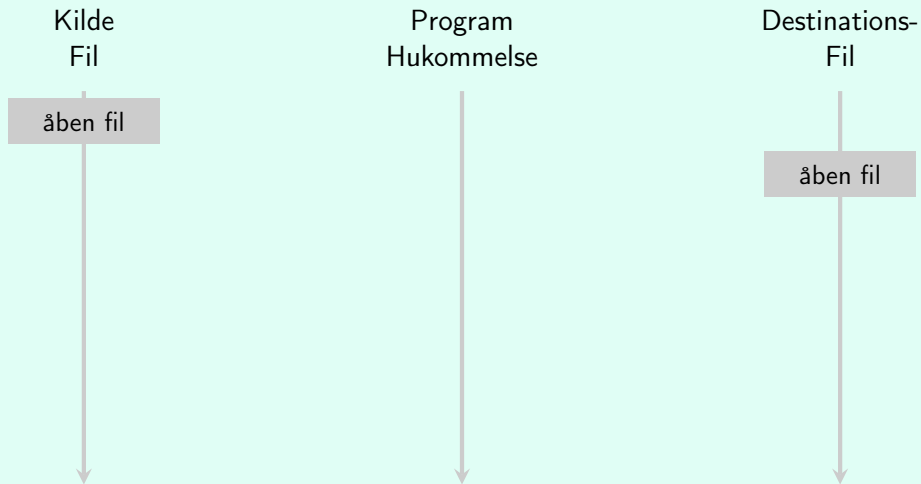
Destinations-
Fil



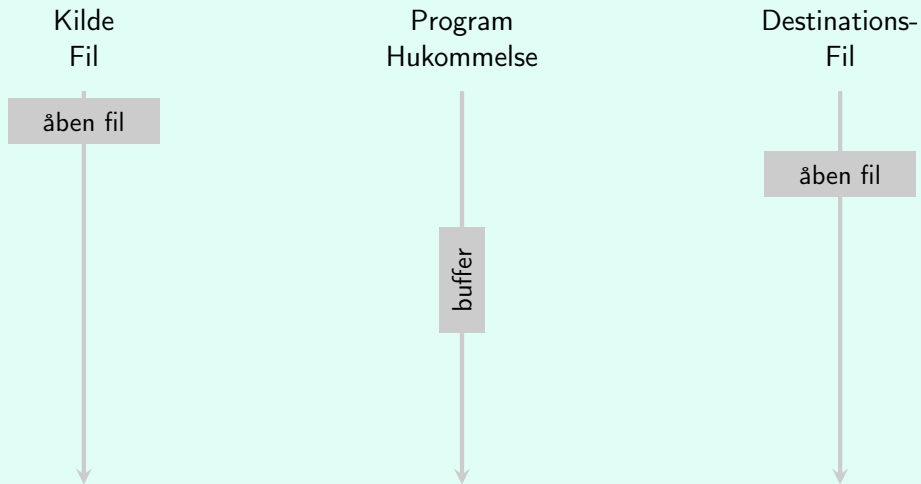
Programmer ▷ Kopiering af Fil



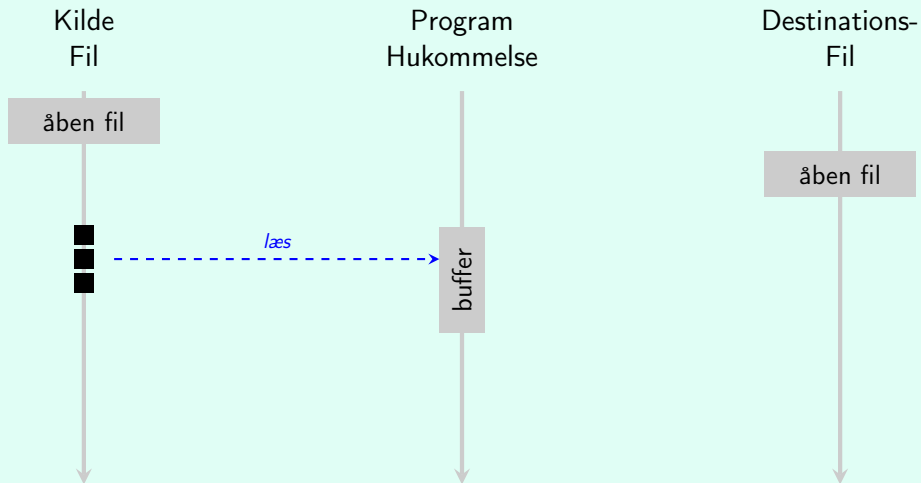
Programmer ▷ Kopiering af Fil



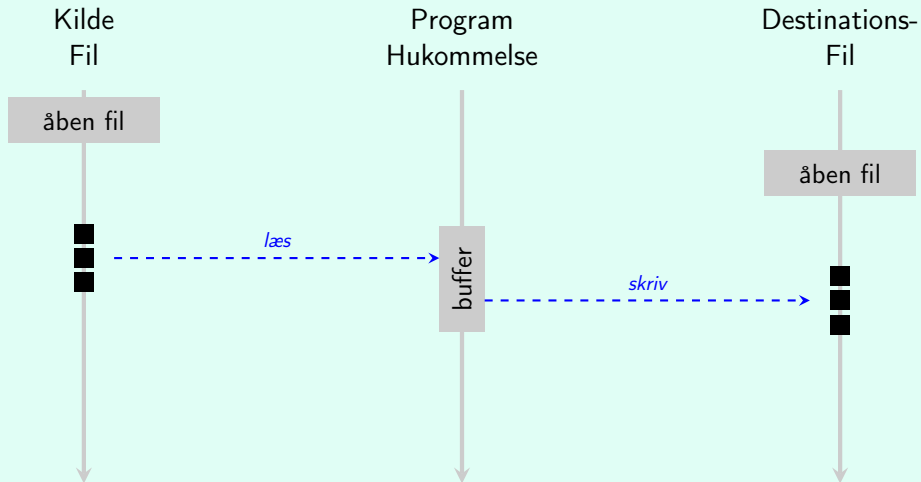
Programmer ▷ Kopiering af Fil



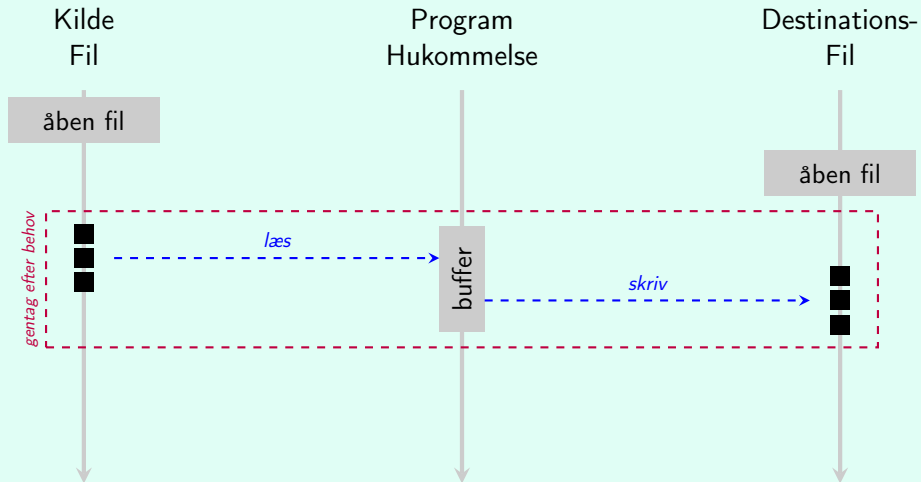
Programmer ▷ Kopiering af Fil



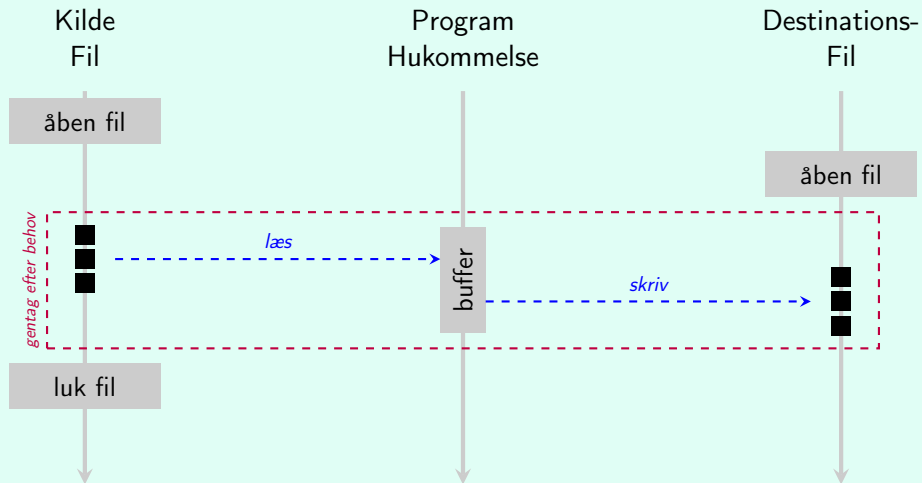
Programmer ▷ Kopiering af Fil



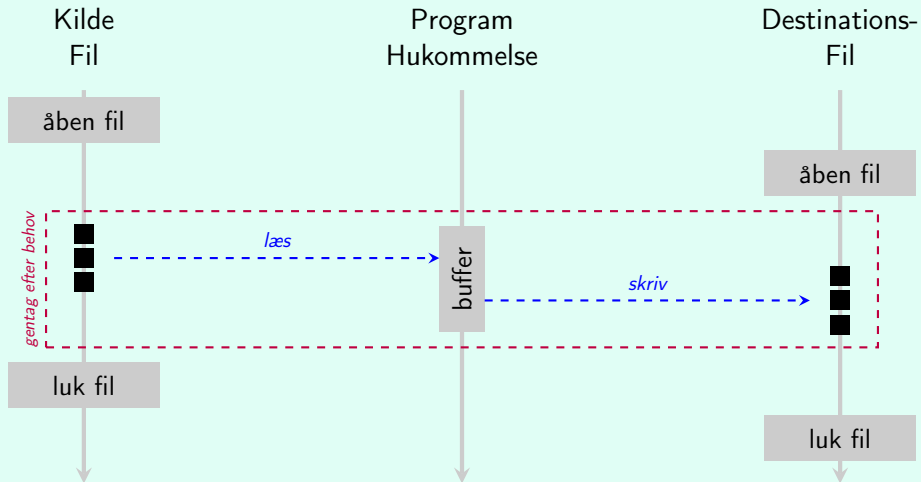
Programmer ▷ Kopiering af Fil



Programmer ▷ Kopiering af Fil



Programmer ▷ Kopiering af Fil



Programmer ► Editering af Fil

Programmer ▷ Editering af Fil

Bruger-
Grænseflade



Programmer ▷ Editering af Fil

Bruger-
Grænseflade



Interessant
Fil



Programmer ▷ Editering af Fil

Bruger-
Grænseflade



Program
Hukommelse



Interessant
Fil



Programmer ▷ Editering af Fil

Bruger-
Grænseflade



Program
Hukommelse



Interessant
Fil

åben fil



Programmer ▷ Editering af Fil

Bruger-
Grænseflade



Program
Hukommelse



Interessant
Fil



Programmer ▷ Editering af Fil

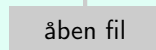
Bruger-
Grænseflade



Program
Hukommelse



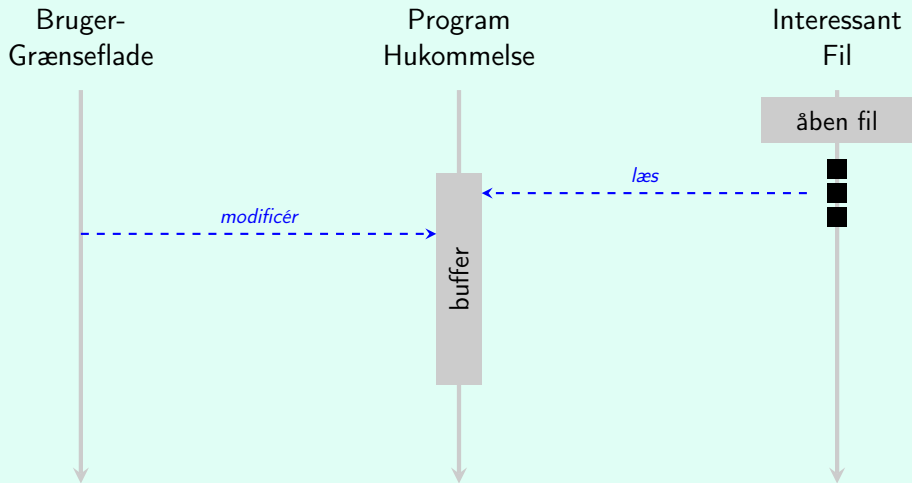
Interessant
Fil



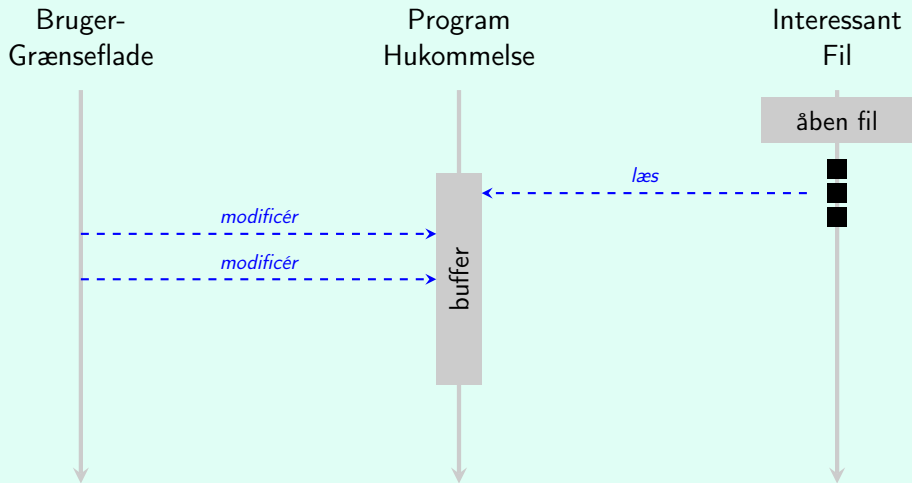
læs



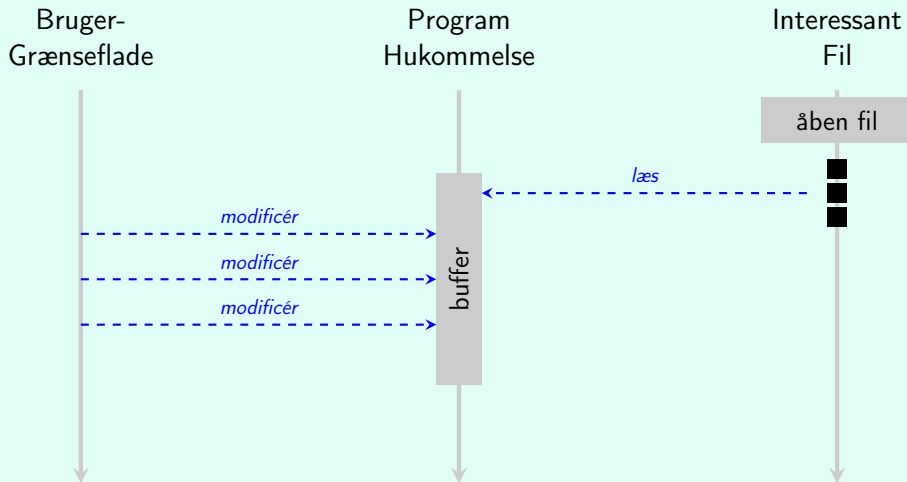
Programmer ▷ Editering af Fil



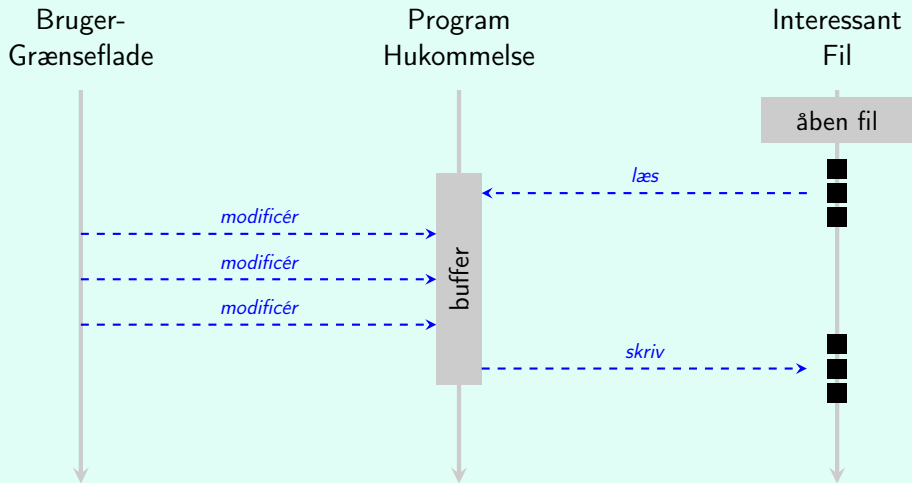
Programmer ▷ Editering af Fil



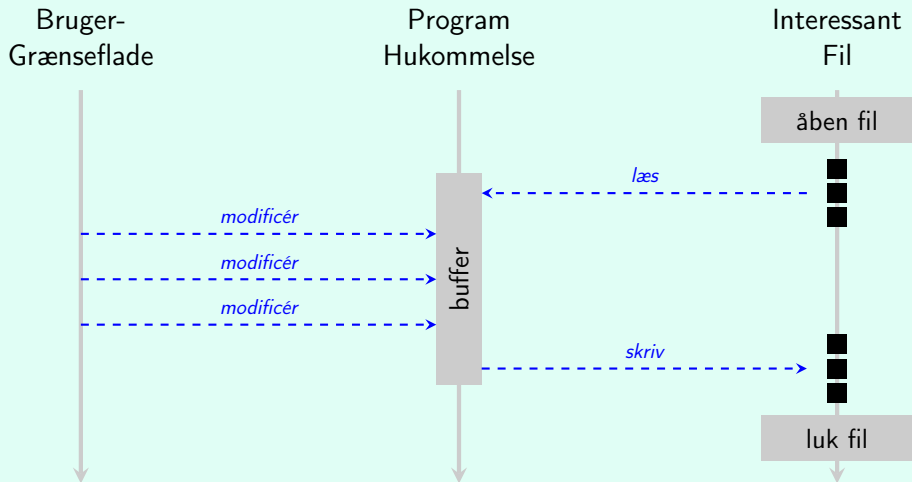
Programmer ▷ Editering af Fil



Programmer ▷ Editering af Fil



Programmer ▷ Editering af Fil



Programmer ▷ Filformater

Programmer ▷ Filformater

Microsoft Word

Microsoft Word 365

LibreWriter

Google Docs

L^AT_EX

Microsoft PowerPoint

Microsoft PowerPoint 365

Programmer ▷ Filformater

Microsoft Word

XML-baseret Word (.docx)

Microsoft Word 365

Traditionel Word (.doc)

LibreWriter

Open Document Format (.odt)

Google Docs

Portable Document Format (.pdf)

L^AT_EX

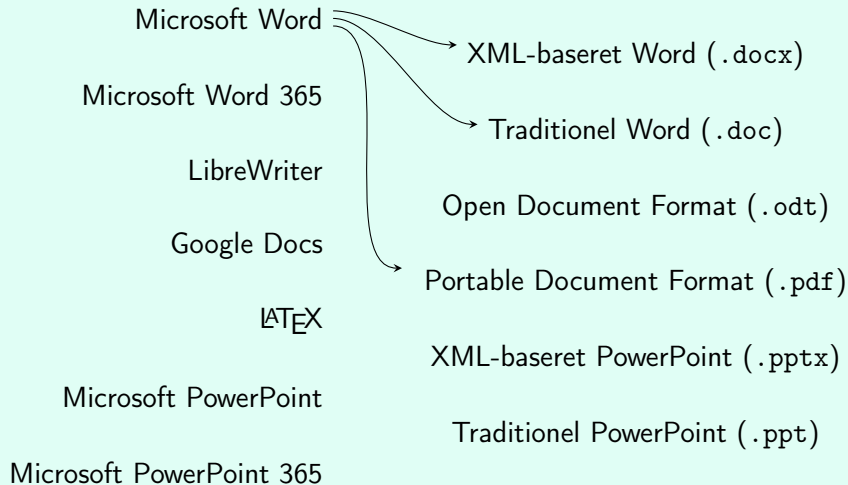
XML-baseret PowerPoint (.pptx)

Microsoft PowerPoint

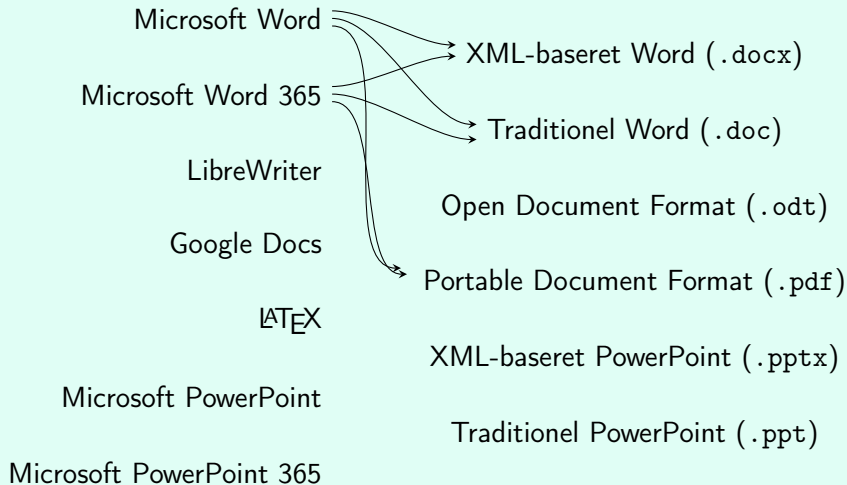
Traditionel PowerPoint (.ppt)

Microsoft PowerPoint 365

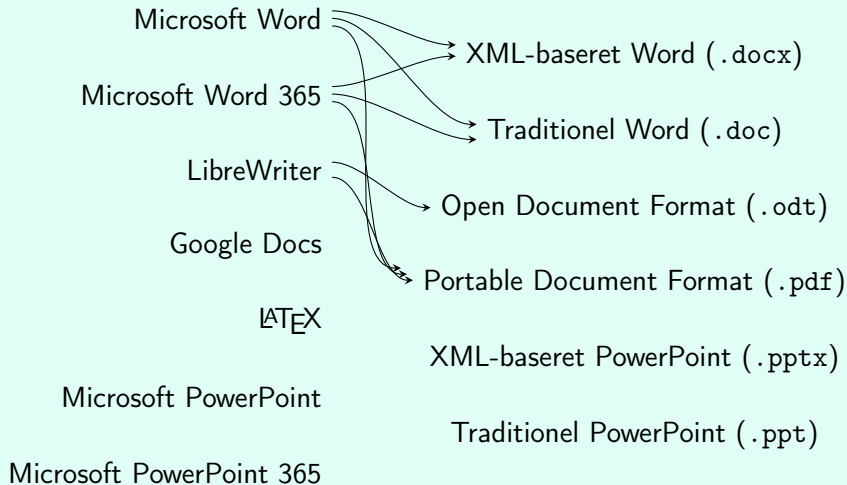
Programmer ▷ Filformater



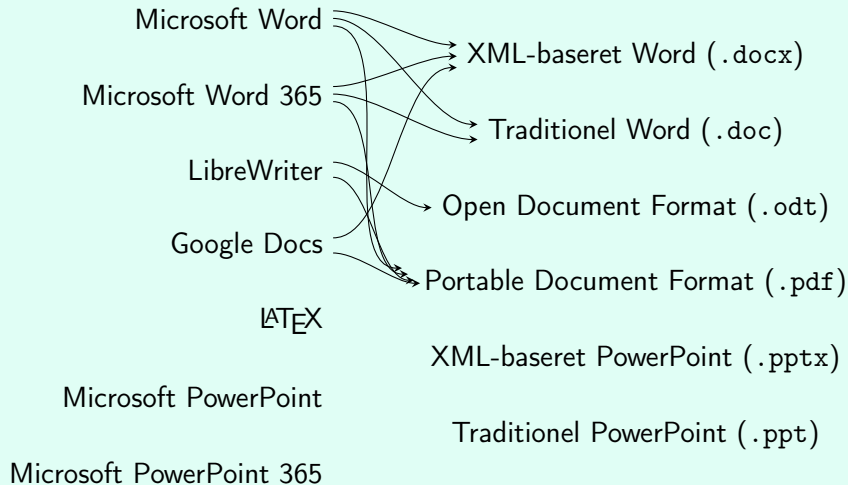
Programmer ▷ Filformater



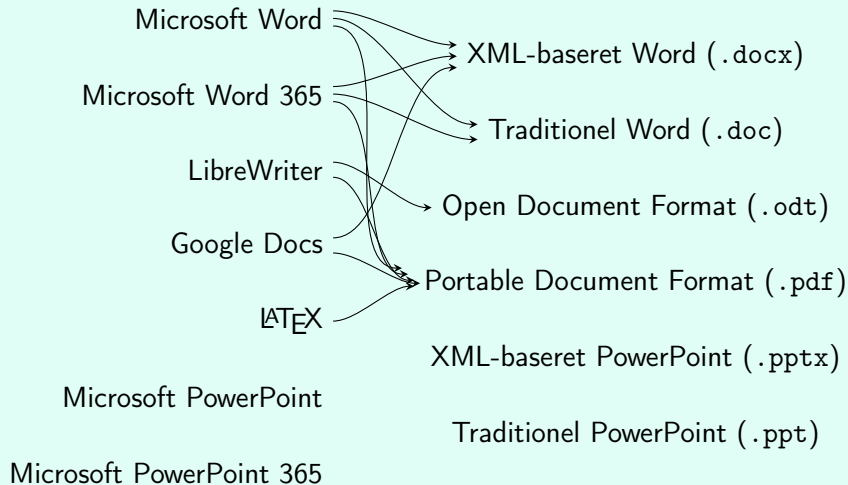
Programmer ▷ Filformater



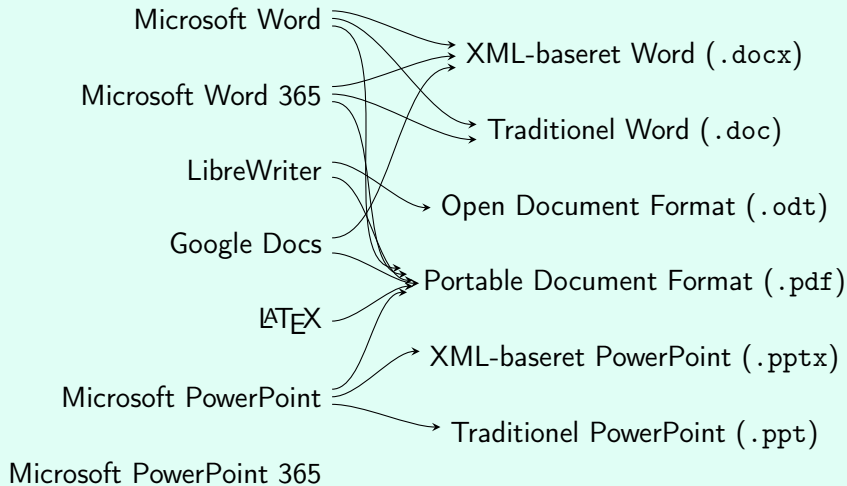
Programmer ▷ Filformater



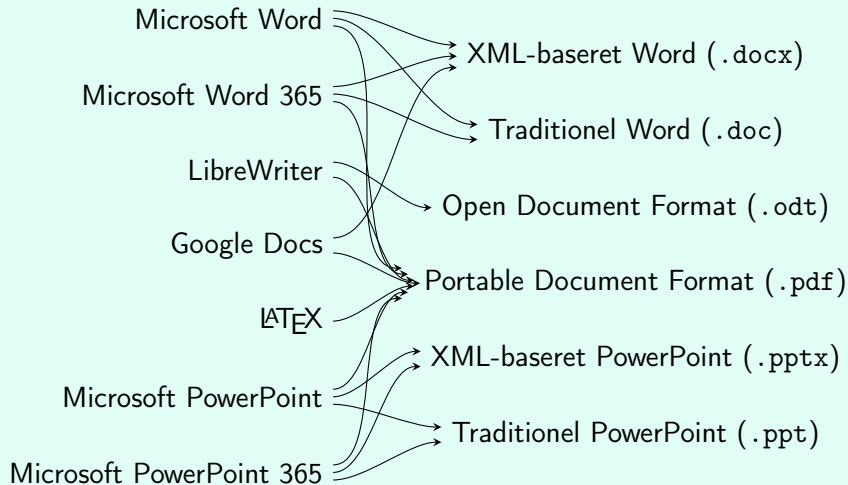
Programmer ▷ Filformater



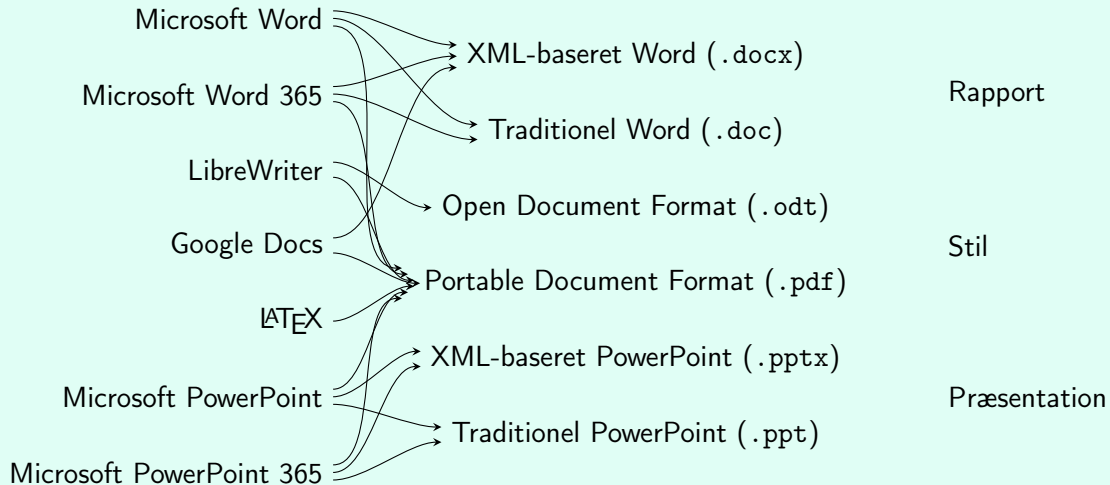
Programmer ▷ Filformater



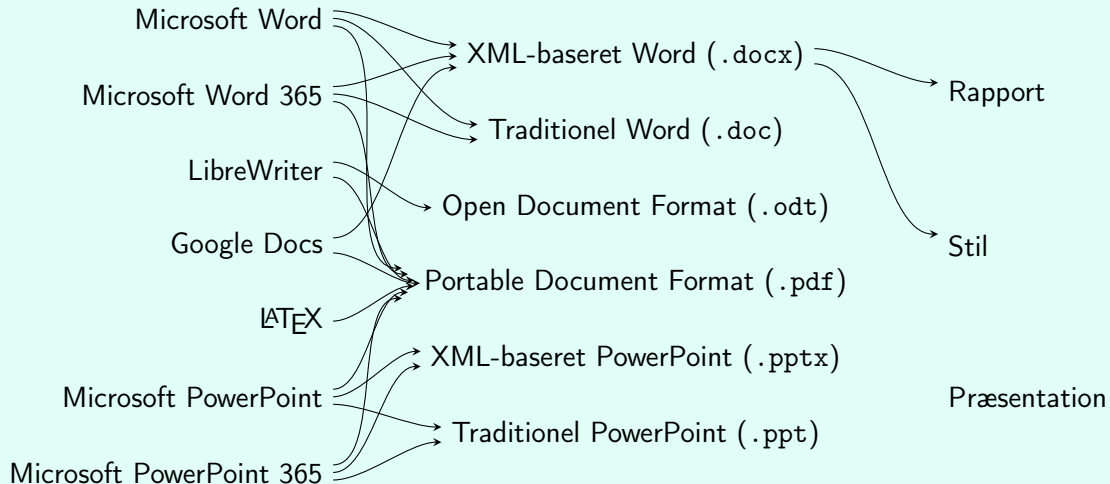
Programmer ▷ Filformater



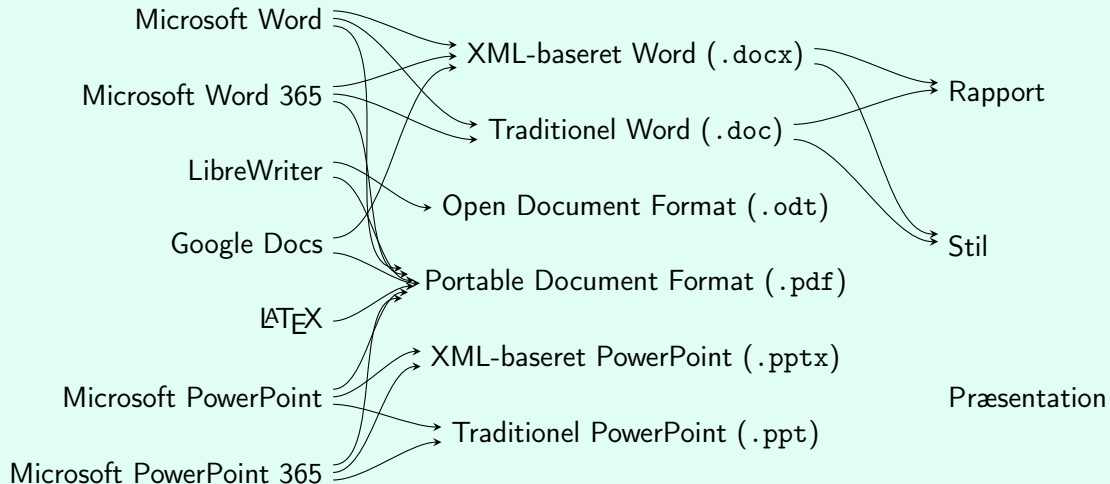
Programmer ▷ Filformater



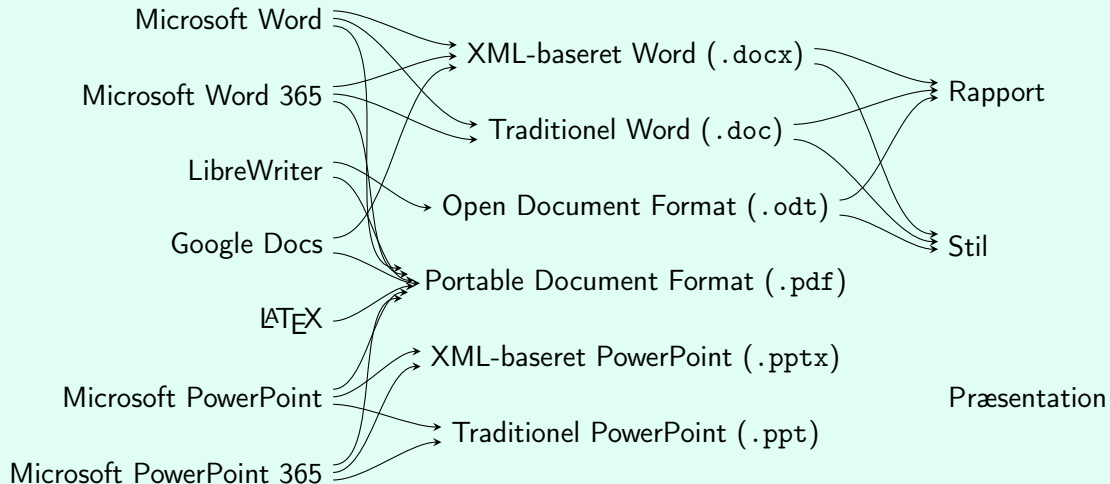
Programmer ▷ Filformater



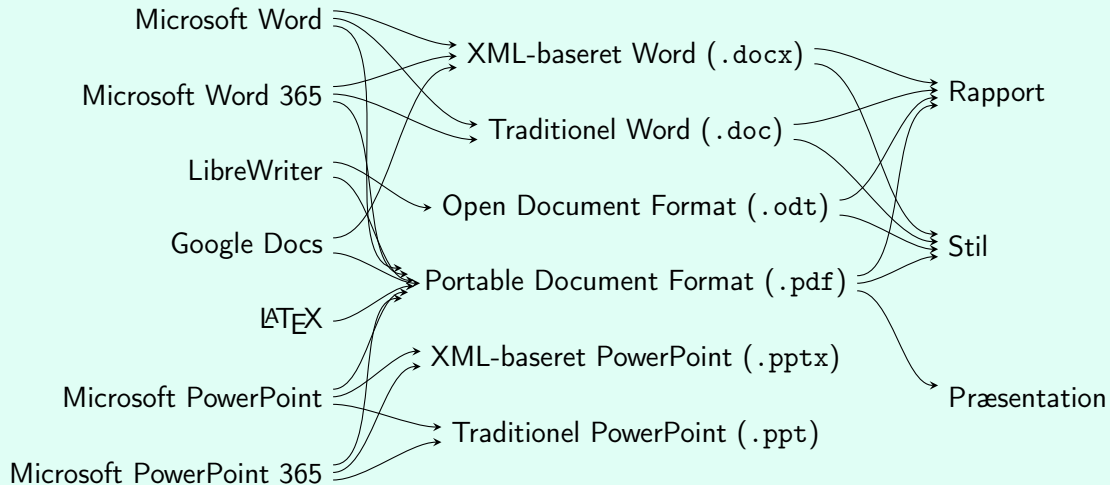
Programmer ▷ Filformater



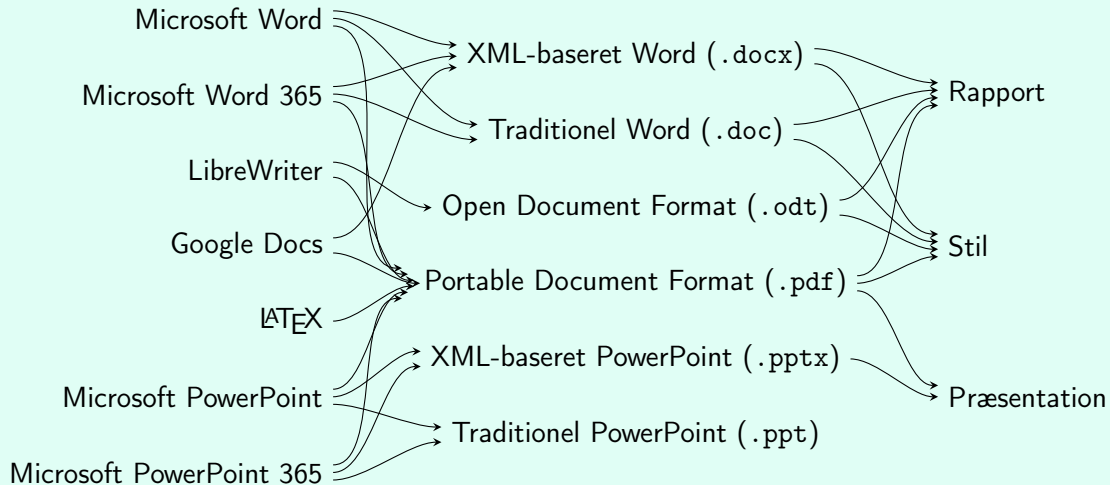
Programmer ▷ Filformater



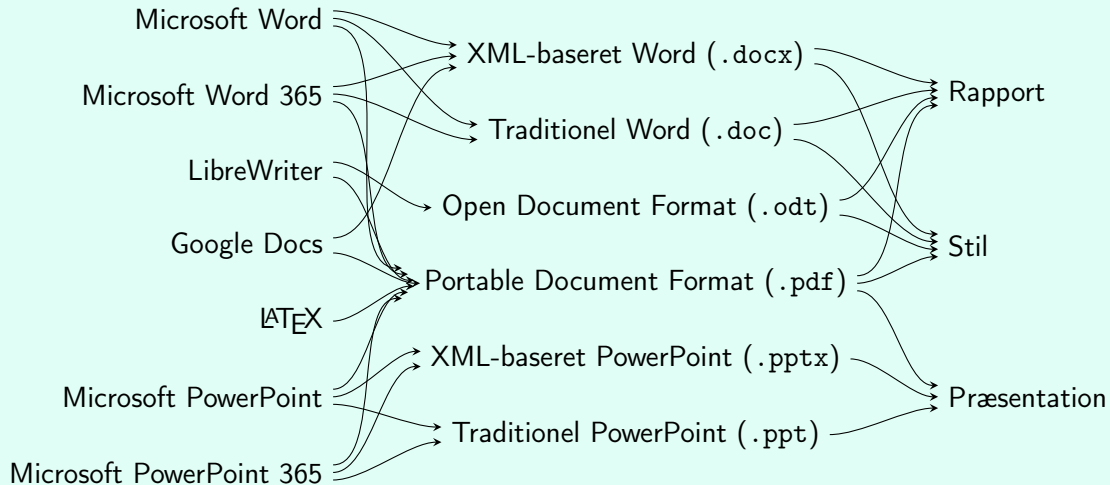
Programmer ▷ Filformater



Programmer ▷ Filformater



Programmer ▷ Filformater



Part 4:

Kommandolinjen

Shells

Terminalen/konsollen giver adgang til en *shell*. Dette er en form for program der giver én mulighed for ved hjælp af kommandoer at starte og stoppe programmer.

Shells

Terminalen/konsollen giver adgang til en *shell*. Dette er en form for program der giver én mulighed for ved hjælp af kommandoer at starte og stoppe programmer.

Nogle shells tilbyder desuden specielle kommandoer til at gøre ens live mere bekvemt. Eksempelvis har de fleste shells en `alias` kommando som man kan bruge til at give en lang og kompliceret kommando et kort navn.

Shells

Terminalen/konsollen giver adgang til en *shell*. Dette er en form for program der giver én mulighed for ved hjælp af kommandoer at starte og stoppe programmer.

Nogle shells tilbyder desuden specielle kommandoer til at gøre ens live mere bekvemt. Eksempelvis har de fleste shells en `alias` kommando som man kan bruge til at give en lang og kompliceret kommando et kort navn.

I en shell udfører man som udgangspunkt én kommando af gangen. Moderne shells holder en historik af de kommandoer man har givet den, og man kan bruge pil op/ned til at genkalde dem, og eventuelt rette dem til. **Brug denne feature.**

Kommandoer

Kommandoer er (typisk små) programmer. De er parameteriserede, så de kan anvendes på mange forskellige typer problemer. Og de kan kombineres.

Kommandoer

Kommandoer er (typisk små) programmer. De er parameteriserede, så de kan anvendes på mange forskellige typer problemer. Og de kan kombineres.

De fleste kommandoer kan kaldes med `--help` som parameter for at få en beskrivelse af hvordan de kan kaldes.

Kommandoer

Kommandoer er (typisk små) programmer. De er parameteriserede, så de kan anvendes på mange forskellige typer problemer. Og de kan kombineres.

De fleste kommandoer kan kaldes med `--help` som parameter for at få en beskrivelse af hvordan de kan kaldes.

Output fra én kommando kan sendes til en anden ved hjælp af `|` (udtales "pipe") operatoren, eller gemmes i en fil med `>` operatoren.

Hyppige Kommandoer

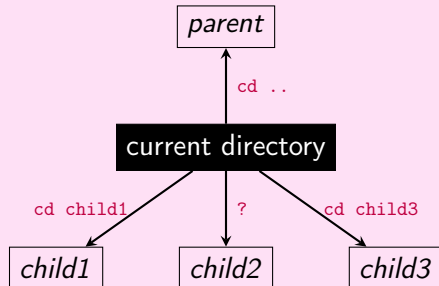
- ▶ `find` kan bruges til at finde alle de stier der matcher et mønster.
- ▶ `grep` vil finde de linjer der matcher et bestemt mønster.
- ▶ `ls` lister indholdet af en directory.
- ▶ `touch` opretter en tom fil.
- ▶ `mkdir` opretter en directory.
- ▶ `rm` sletter en fil.
- ▶ `mv` flytter en fil.
- ▶ `nano` er en af mange tekst editorer. Når programmet er aktivt vil man i bunden kunne se nogle shortcuts til nogle hyppige operationer. Her er `^` en shorthand for `Ctrl`.
- ▶ `ps` lister kørende processer.

Navigation af Filsystem

For at ændre *current directory*, bruges kommandoen `cd` (*change directory*). Den tager én parameter som er en retning. Retningen er enten navngivet `..` (mod træets rod), eller efter barne-knuden i træet.

Navigation af Filsystem

For at ændre *current directory*, bruges kommandoen `cd` (*change directory*). Den tager én parameter som er en retning. Retningen er enten navngivet `..` (mod træets rod), eller efter barne-knuden i træet.



Listning af Directory

Indholdet af en directory kan listes med kommandoen `ls` (*list*).

Listning af Directory

Indholdet af en directory kan listes med kommandoen `ls` (*list*).

Den tager en lang række parametre. Ofte anvendte varianter:

- ▶ *Få et hurtigt overblik:* `ls`
- ▶ *Få flere informationer:* `ls -l`
- ▶ *Finde de sidst modificerede filer:* `ls -ltr`
- ▶ *Finde de største:* `ls -lSr`

Listning af Directory

Indholdet af en directory kan listes med kommandoen `ls` (*list*).

Den tager en lang række parametre. Ofte anvendte varianter:

- ▶ *Få et hurtigt overblik:* `ls`
- ▶ *Få flere informationer:* `ls -l`
- ▶ *Finde de sidst modificerede filer:* `ls -ltr`
- ▶ *Finde de største:* `ls -lSr`

Bemærk: En computer vil se forskel på store og små bogstaver. Nogle filsystemer er designet til at se bort fra dette, men det er ikke problemfrit.

Part 5: Styresystemer

Programmer

En moderne processor har flere *kerner* (eng: **core**). De kan hver arbejde på ét program af gangen.

Programmer

En moderne processor har flere *kerner* (eng: **core**). De kan hver arbejde på ét program af gangen.

Et styresystem har en *kerne* (eng: **kernel**) der holder styr på de programmer der er *loaded ind* i hukommelsen. Nogle af disse programmer er *passive* (de venter på input), og andre er *aktive*. Kernen (**kernel**) sørger for at give de aktive programmer tid på en kerne (**core**). Kernen **kernel** bruger en kø til dette, og når et program har fået lidt tid pliver den placeret bagerst i køen.

Programmer

En moderne processor har flere *kerner* (eng: **core**). De kan hver arbejde på ét program af gangen.

Et styresystem har en *kerne* (eng: **kernel**) der holder styr på de programmer der er *loaded ind* i hukommelsen. Nogle af disse programmer er *passive* (de venter på input), og andre er *aktive*. Kernen (**kernel**) sørger for at give de aktive programmer tid på en kerne (**core**). Kernen **kernel** bruger en kø til dette, og når et program har fået lidt tid pliver den placeret bagerst i køen.

På alle computere kører der et hav af programmer. Nogle få af dem har en grafisk grænseflade, og de kan godt være aktive selv om de ikke kan ses på skærmen. Andre programmer yder services til disse programmer.

Programmer

En moderne processor har flere *kerner* (eng: **core**). De kan hver arbejde på ét program af gangen.

Et styresystem har en *kerne* (eng: **kernel**) der holder styr på de programmer der er *loaded ind* i hukommelsen. Nogle af disse programmer er *passive* (de venter på input), og andre er *aktive*. Kernen (**kernel**) sørger for at give de aktive programmer tid på en kerne (**core**). Kernen **kernel** bruger en kø til dette, og når et program har fået lidt tid pliver den placeret bagerst i køen.

På alle computere kører der et hav af programmer. Nogle få af dem har en grafisk grænseflade, og de kan godt være aktive selv om de ikke kan ses på skærmen. Andre programmer yder services til disse programmer.

Mens Aslak skriver dette kører der 500+ programmer på hans laptop.

Eksamenssituationer

Hvis I kun må have nogle bestemte programmer åbne, så må andre programmer **ikke** være loaded. Det betyder blandt andet at det ikke må kunne ses i styresystemets overblik.

Eksamenssituationer

Hvis I kun må have nogle bestemte programmer åbne, så må andre programmer **ikke** være loaded. Det betyder blandt andet at det ikke må kunne ses i styresystemets overblik.

Nogle programmer har flere funktionaliteter. Eksempelvis er der PDF læsere med “indbygget ChatGPT”.

Eksamenssituationer

Hvis I kun må have nogle bestemte programmer åbne, så må andre programmer **ikke** være loaded. Det betyder blandt andet at det ikke må kunne ses i styresystemets overblik.

Nogle programmer har flere funktionaliteter. Eksempelvis er der PDF læsere med "indbygget ChatGPT".

Hvis I står i en testsituation hvor ChatGPT ikke må bruges, men I godt må bruge en PDF læser, så må PDF læsere med denne funktionalitet **ikke** bruges.

Questions?

