COS
Computersystemer

Lektion #6
Operativ systemer

Morten Hansen
moh@sdu.dk

Pointgivende aktivitet.

?

Morten Hansen
moh@sdu.dk

Social event

Kl.: 13:00

⇕

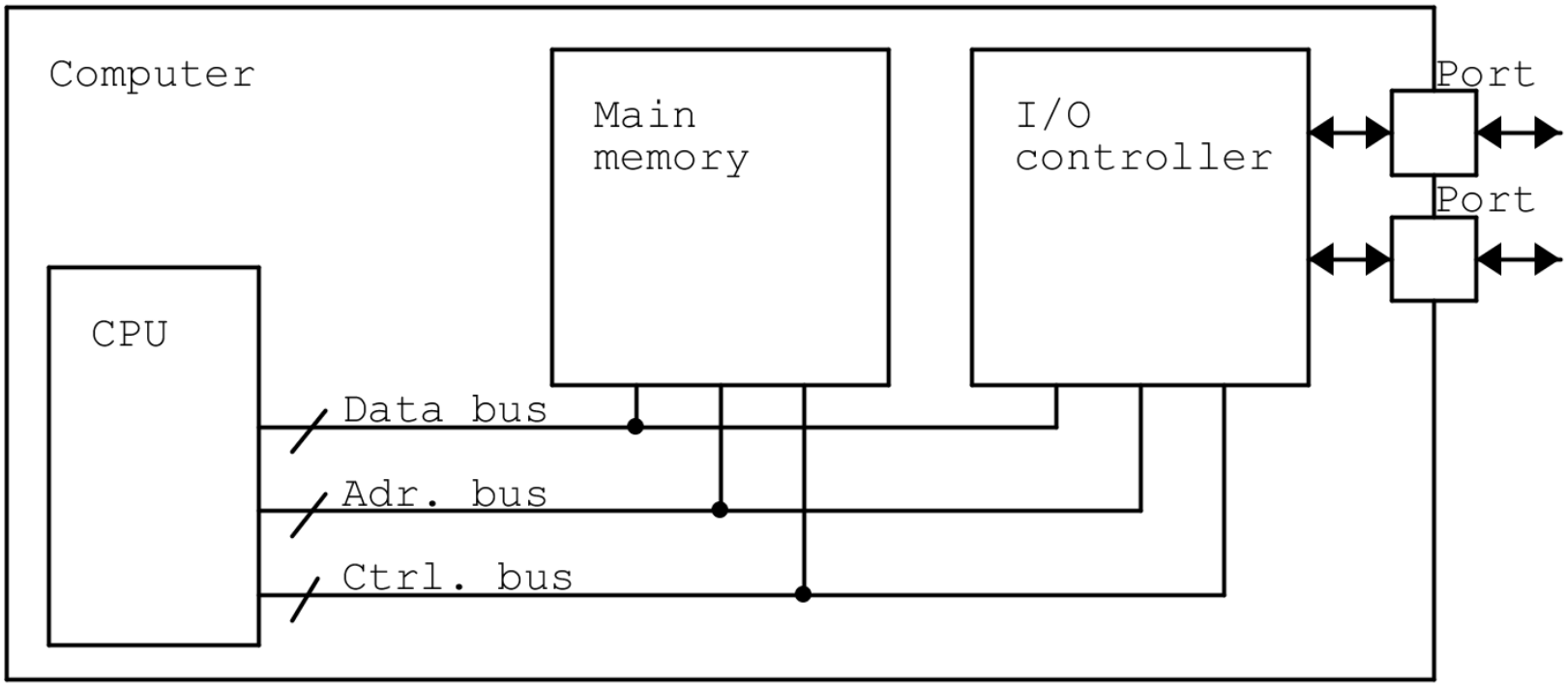
Instruktortime: 12:00-13:00

Morten Hansen
moh@sdu.dk

Eksempel med afvikling af program med I/O.

Morten Hansen
moh@sdu.dk

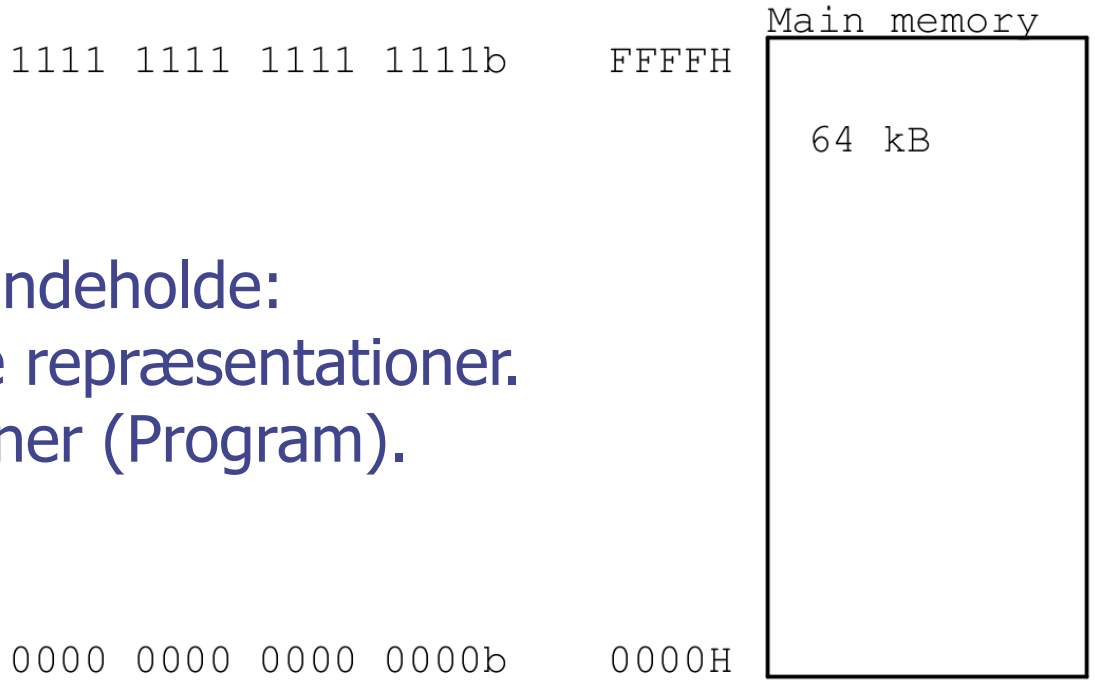
Computer med I/O. (I/O mapped I/O).



Ydre enheder

Morten Hansen
moh@sdu.dk

Main memory.



Main memory kan indeholde:

- Data i forskellige repræsentationer.
- Maskininstruktioner (Program).

Morten Hansen
moh@sdu.dk

Programmering

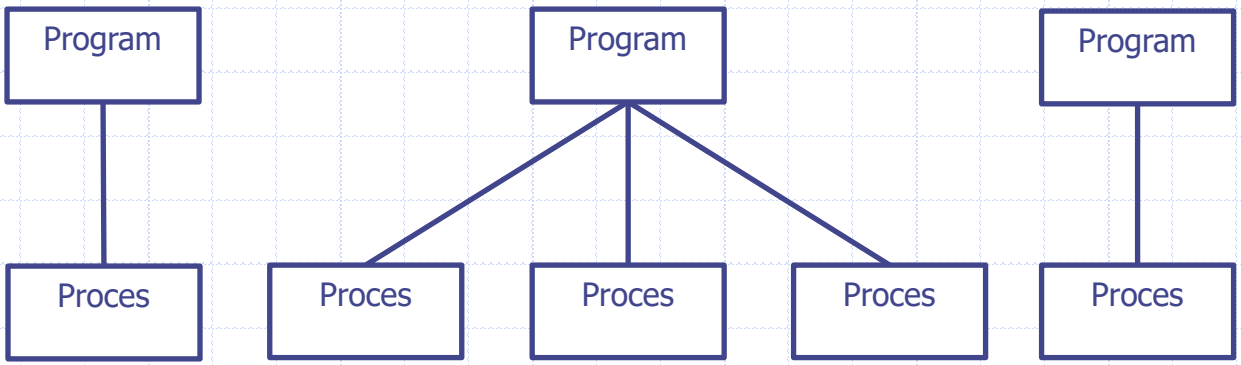
Adder de binære værdier i adr.: 0x40 og 0x41. Aflever resultatet i adr.: 0x42

```
Maskin-          Assembler                  Høj niveau kode
instruktioner    Kode                       3. generations
                                            programmeringssprog
                                            (her: C)
-------------------------------------------------------------------------
0x1140           LOAD  R1,[0x40]            Resultat = Addent1 + Addent2;
0x1241           LOAD  R2,[0x41]
0x5012           ADD   R0, R1, R2
0x3042           STORE R0,[0x42]
```


Morten Hansen
moh@sdu.dk

Program vs. proces

En proces er et program under afvikling på en computer.



Flere processer kan afvikle den samme kode!

Morten Hansen
moh@sdu.dk

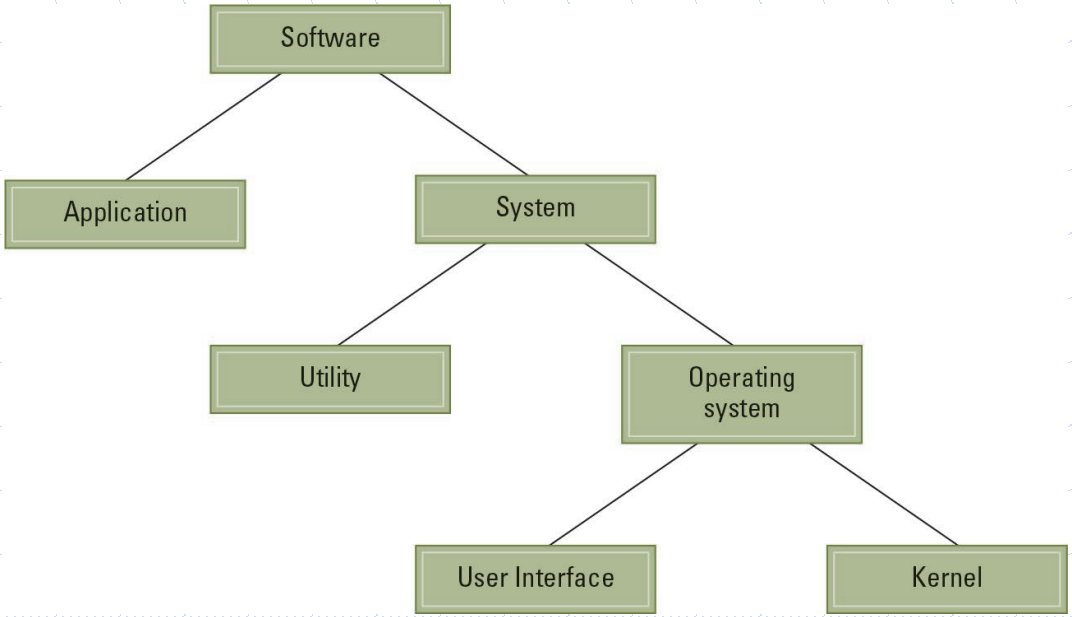
Programmering

Adder de binære værdier i adr.: 0x40 og 0x41. Aflever resultatet i adr.: 0x42

```
Maskin-          Assembler                Høj niveau kode
instruktioner    Kode                     3. generations
                                          programmeringssprog
                                          (her: C)
-------------------------------------------------------------------------
0x1140           LOAD  R1,[0x40]          Resultat = Addent1 + Addent2;
0x1241           LOAD  R2,[0x41]
0x5012           ADD   R0, R1, R2
0x3042           STORE R0,[0x42]
```

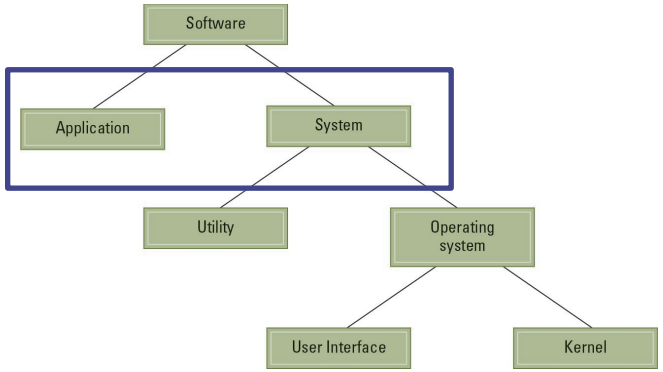

Morten Hansen
moh@sdu.dk

Software klassifikation



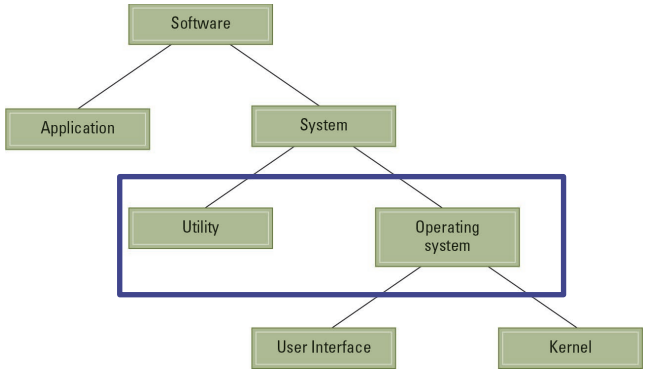
Morten Hansen
moh@sdu.dk

Software classification

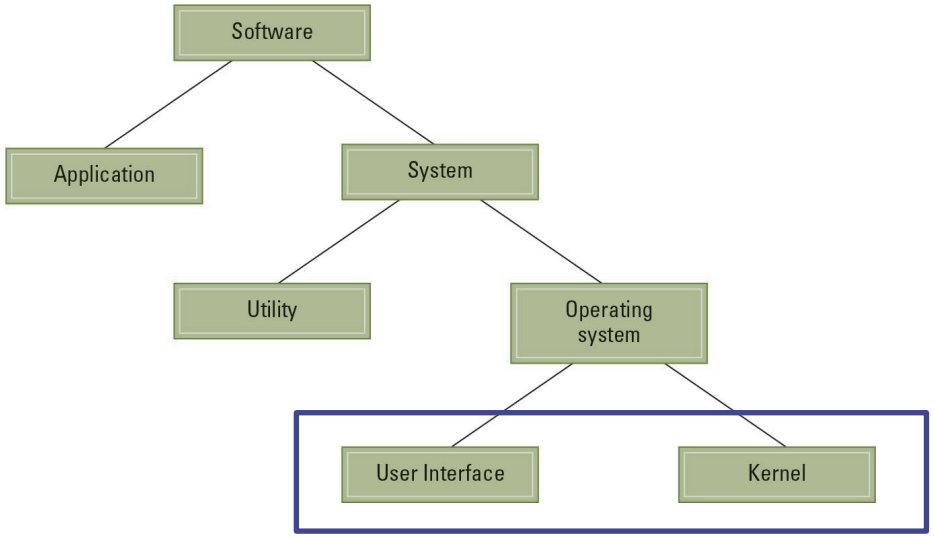


- **Application software**
  - Performs specific tasks for users (productivity, games, software development).
  - De typiske programmer som vi alle kommer til at udvikle.
- **System software**
  - Provides infrastructure for application software.
  - Consists of operating system and utility software.

System classification



- **Utility**
  - Der er ikke nogen klar definition mellem applikation og Utility software.
  - Software som bruges som hjælpe software til selve Operativsystemer
    - Kopiere filer mellem interne ender
- **Operativsystemer**
  - Selve Operativsystemer



Operating System Components

- **User Interface:** Communicates with users
  - Text based (Shell)
  - Graphical user interface (GUI)
- **Kernel:** Performs basic required functions
  - File manager
  - Device drivers
  - Memory manager
  - Scheduler and dispatcher

Hvad kunne vi ønske os af et OS?

Morten Hansen
moh@sdu.dk

Funktionerne udført af operativsystemet

- An operating system coordinates the execution of application software, utility software, and units within the operating system itself.
  - Oversee operation of computer
  - Store and retrieve files
  - Provide the user interface to request execution of programs
  - Coordinate the execution of programs

Morten Hansen
moh@sdu.dk

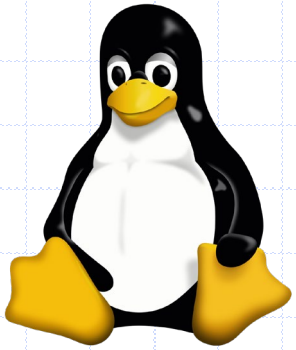
Computer klasser

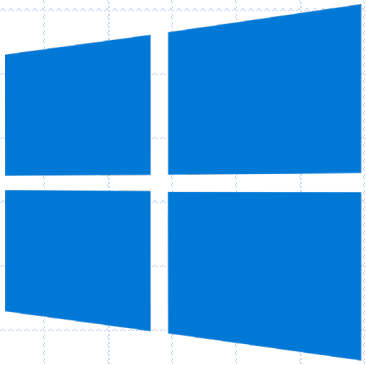
General purpose		Mainframes
		PCer
		Mobil telefoner
Embedded computers	• Formålsbundet • Real-time	Apparater
		”Things”

Morten Hansen
moh@sdu.dk

Kendte Operativ systemer

macOS





- Desktop
  - Windows
  - Mac OS
  - Linux/UNIX
- Smart Phone
  - Apple iOS
  - Google Android
- Embedded
  - FreeRTOS
  - ThreadX
  - µC/OS
  - VxWorks
  - Diverse proprietære operativsystemer

iOS

freeRTOS

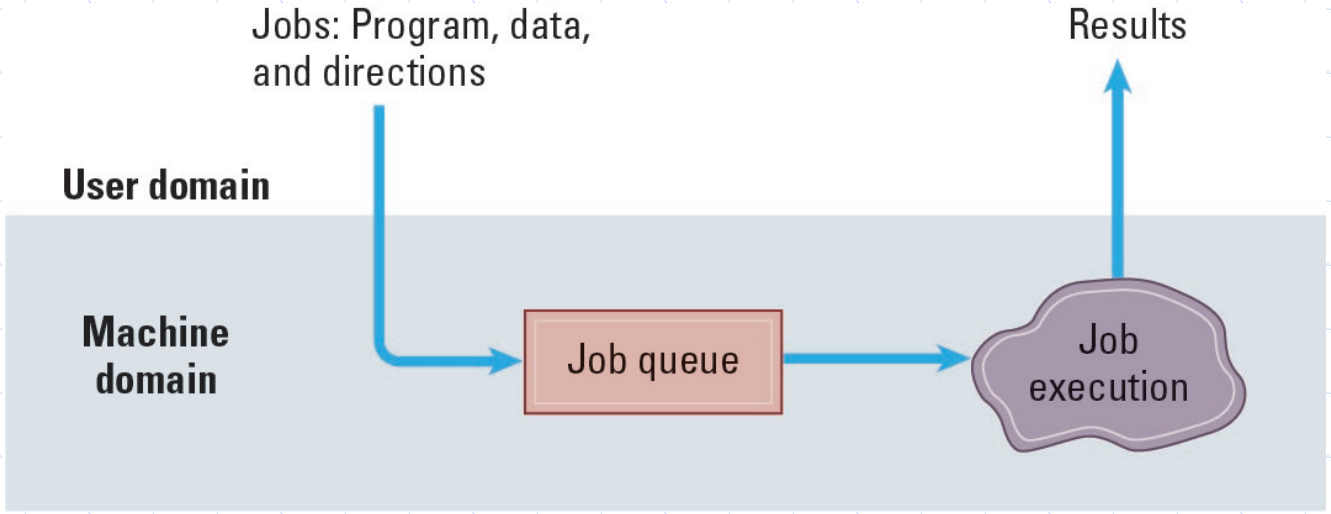
Morten Hansen
moh@sdu.dk

Historien omkring Operativsystem

- Hvert program blev kaldt et “job”
- Tidlige computer krævede en stor del konfiguration.
  - Da hvert job krævede sit engen konfiguration.
  - Operativsystem startede med at være en enklere måde at lave opsætning og overgange mellem ”jobs”.
- Program og Operativsystem paradigmer
  - Batch processing (job queue)
  - Interactive processing (real time)
  - Time-sharing (one machine, many users)
  - Multitasking (one user, many tasks)
  - Multiprocessor machines (load balancing)
  - Embedded Systems (specific devices)

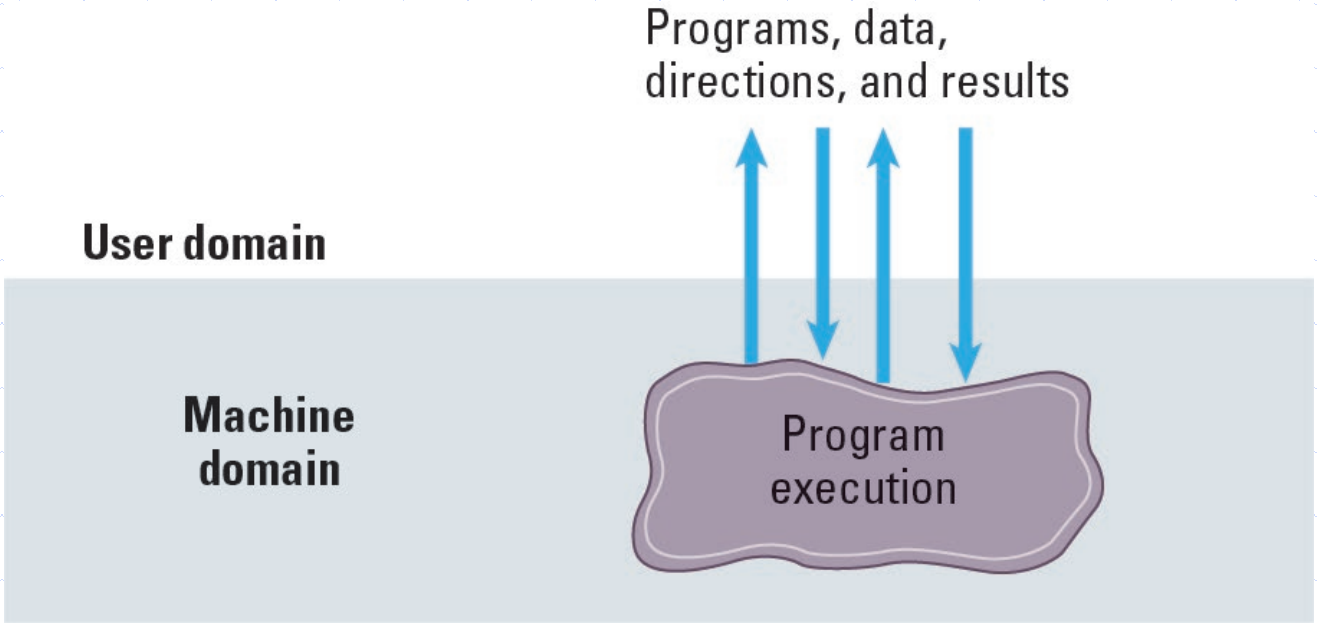
Morten Hansen
moh@sdu.dk

Figure 3.1 Batch processing



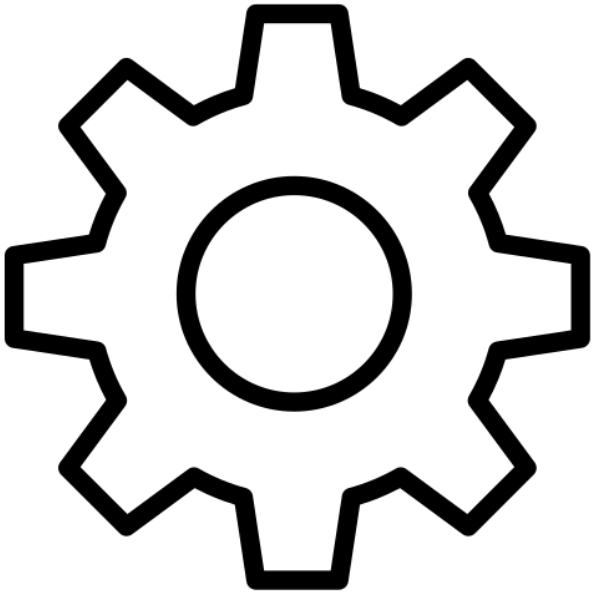
Morten Hansen
moh@sdu.dk

Figure 3.2 Interactive processing



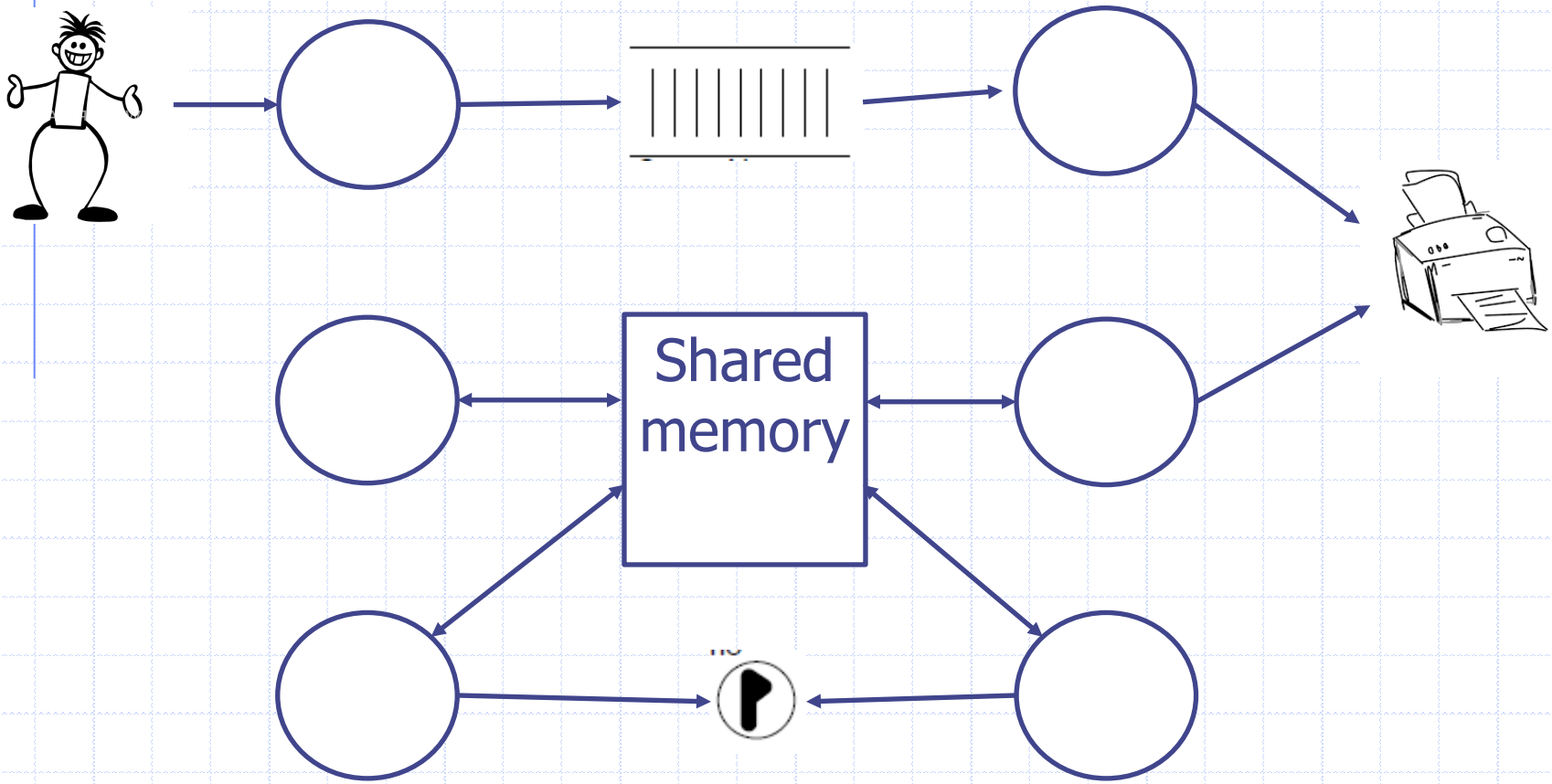
Morten Hansen
moh@sdu.dk

Proces konceptet



- **Proces:**
  - The activity of executing a program.
- **Procesens status:**
  - Processens tilstand (State).
  - Program counter.
  - General purpose registers.
  - Pladser i main memory.
    - Indeholder instruktionerne for processen.

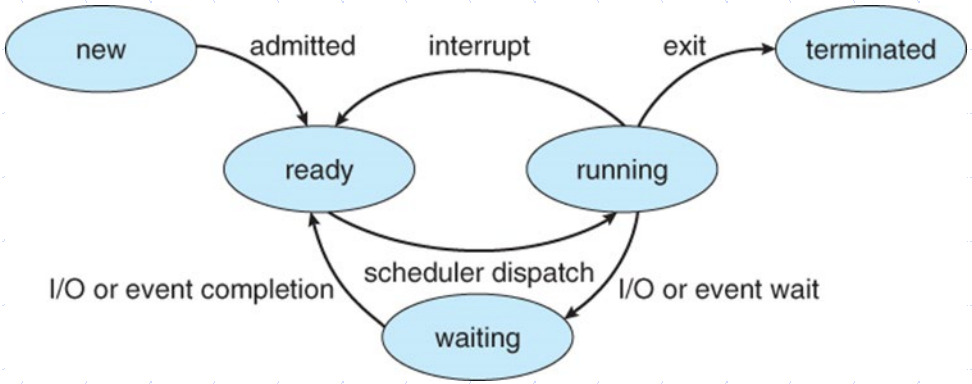
Processdiagram



Morten Hansen
moh@sdu.dk

Process State

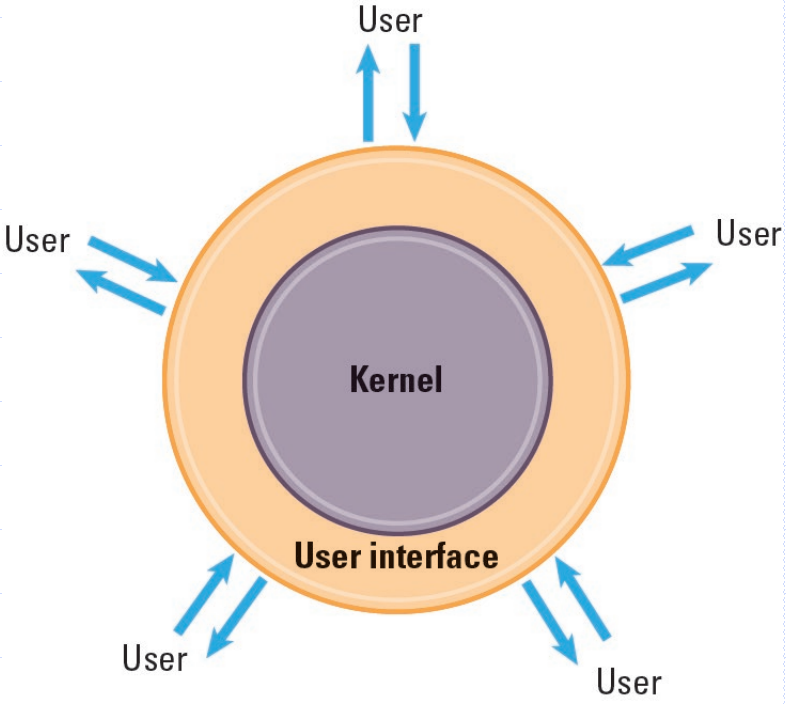
- **New**: Processen er blevet oprettet.
- **Ready**: Processen har alle reusser klar og venter på at få tid på CPU'en.
- **Running**: CPU'en er ved at køre processen.
- **Waiting**: Processen venter på en resurse. F.eks et keyboard input eller en timer.
- **Termainated**: Processen er færdig.



Morten Hansen
moh@sdu.dk

User interface

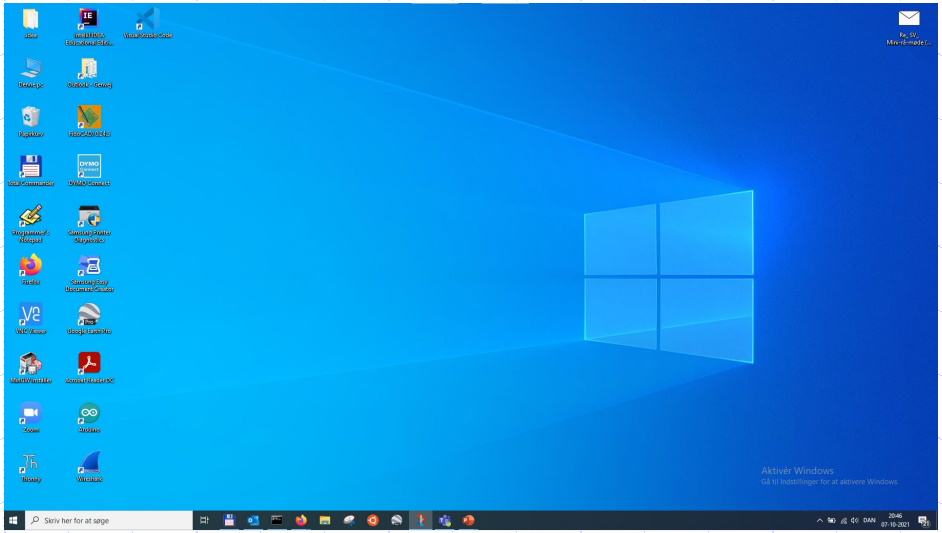
The user interface acts as an intermediary between users and the operating system's kernel



Morten Hansen
moh@sdu.dk

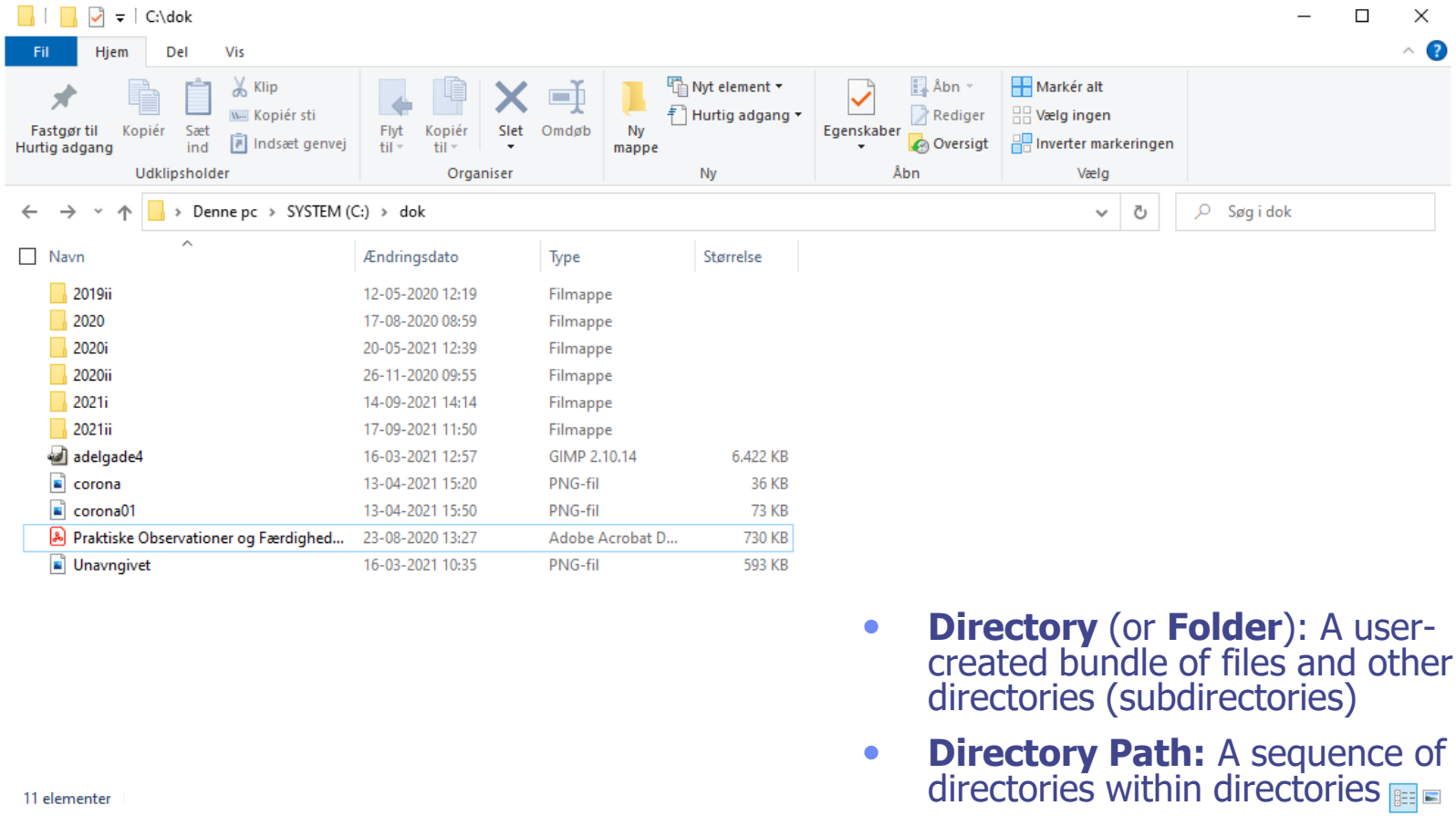
SHELL eller GUI

```
C:\Users\moh>cd ..

C:\Users>cd ..

C:\>dir
 Volume in drive C is SYSTEM
 Volume Serial Number is 26AE-8BCB

 Directory of C:\

25-07-2021  14:05    <DIR>          dok
01-11-2020  10:02    <DIR>          Intel
05-10-2021  20:55    <DIR>          lib
23-07-2021  10:48    <DIR>          lnk
30-08-2021  18:16    <DIR>          Ny mappe
01-11-2020  09:39    <DIR>          PerfLogs
02-11-2020  12:23    <DIR>          prg
30-09-2021  07:56    <DIR>          Program Files
17-09-2021  21:24    <DIR>          Program Files (x86)
17-09-2021  22:45    <DIR>          projekt
22-01-2021  18:21    <DIR>          raw
21-12-2020  17:30    <DIR>          set
04-10-2021  12:28    <DIR>          tmp
23-07-2021  09:41    <DIR>          Users
26-09-2021  11:25    <DIR>          Windows
               0 File(s)              0 bytes
              15 Dir(s)  174.904.205.312 bytes free

C:\>
```



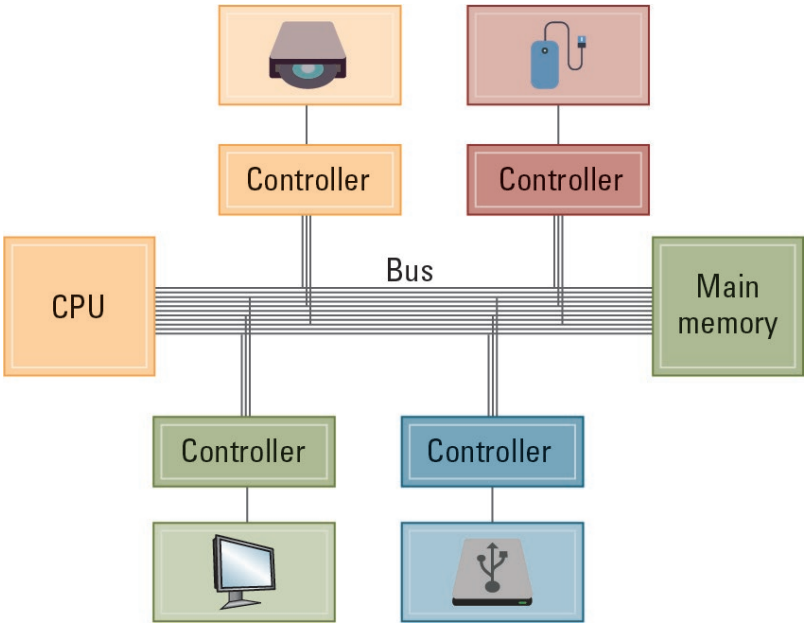
Morten Hansen
moh@sdu.dk

File Manager



- **Directory** (or **Folder**): A user-created bundle of files and other directories (subdirectories)
- **Directory Path:** A sequence of directories within directories
- Ansvarlig for hvilke burger der har adgang til hvilke filer.

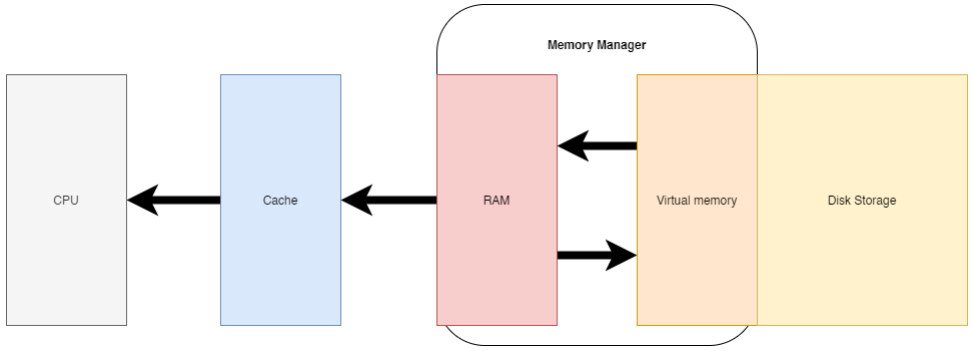
SDU



Device drivers

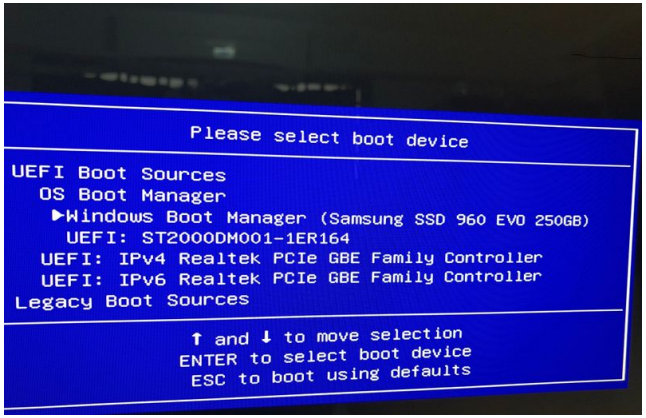
- System i kernen, som er ansvarlig for at ”oversætte” input fra controllerne.
- Hver driver er unik til det enkelte eksterne enhed.

Memory Manager



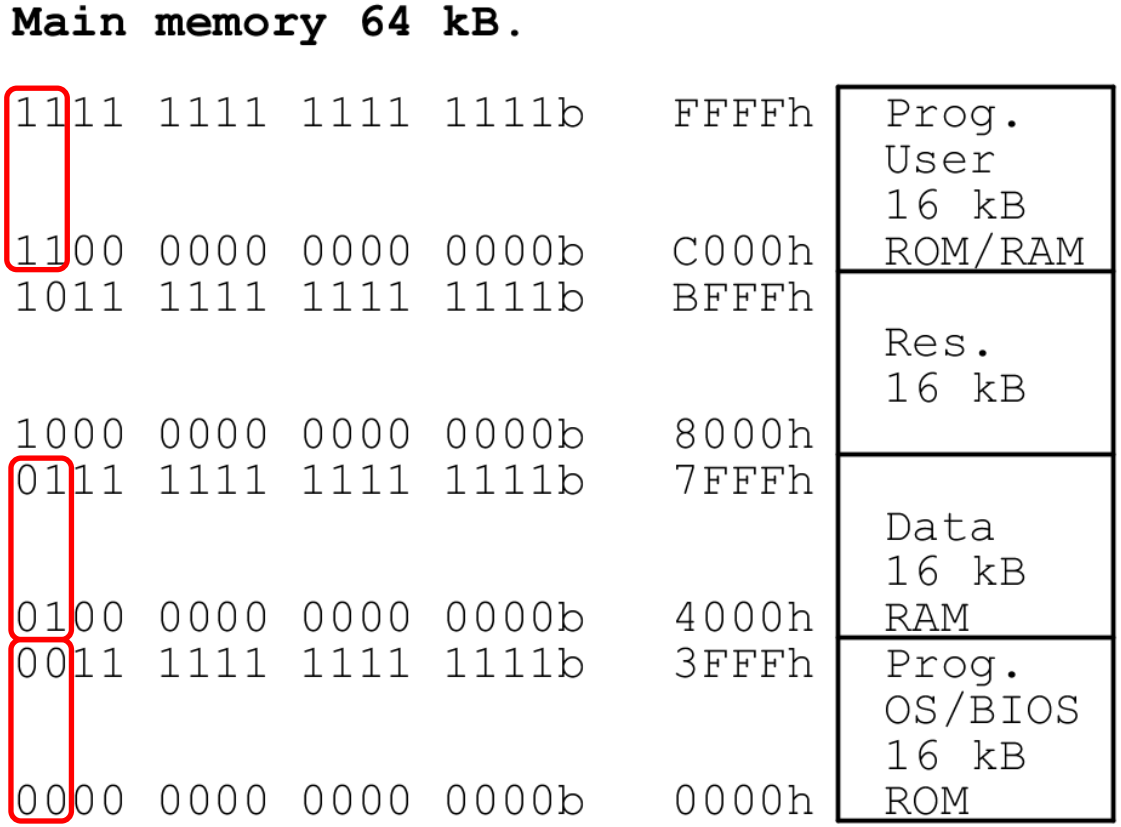
- Allocates space in main memory
- May create the illusion that the machine has more memory than it does (**virtual memory**) by playing a "shell game" in which blocks of data (**pages**) are shifted back and forth between main memory and mass storage.

Bootstrap



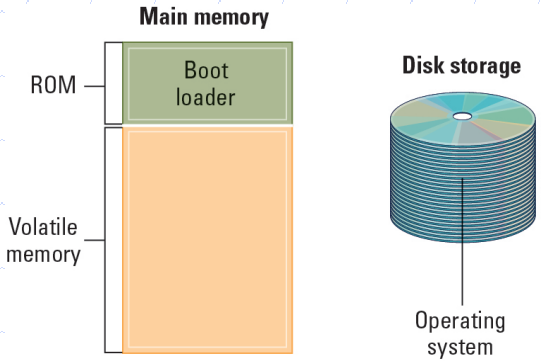
- **Boot loader:** Program in ROM (example of firmware)
  - Run by the CPU when power is turned on
  - Transfers operating system from mass storage to main memory
  - Executes jump to operating system

Memory Map.

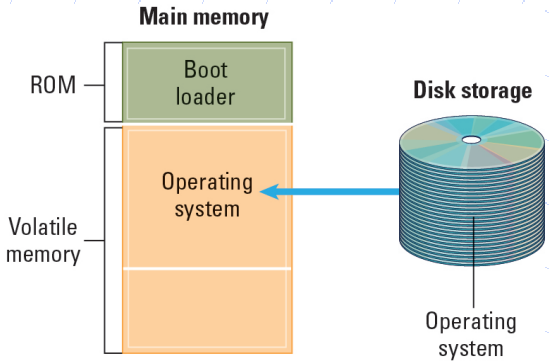


Morten Hansen
moh@sdu.dk

The booting process



Step 1: Machine starts by executing the boot loader program already in memory. Operating system is stored in mass storage.



Step 2: Boot loader program directs the transfer of the operating system into main memory and then transfers control to it.

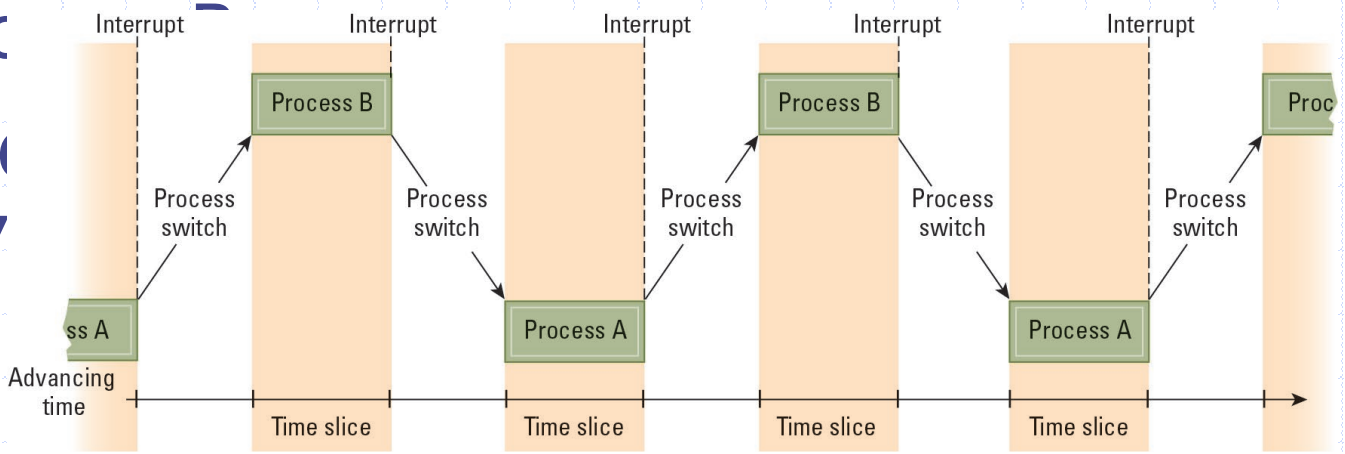
Morten Hansen
moh@sdu.dk

Scheduler and dispatcher

- **Scheduler:**
  - Adds new processes to the process table and removes completed processes from the process table.
- **Dispatcher:**
  - Controls the allocation of time slices to the processes in the process table
    - The end of a time slice is signaled by an interrupt.

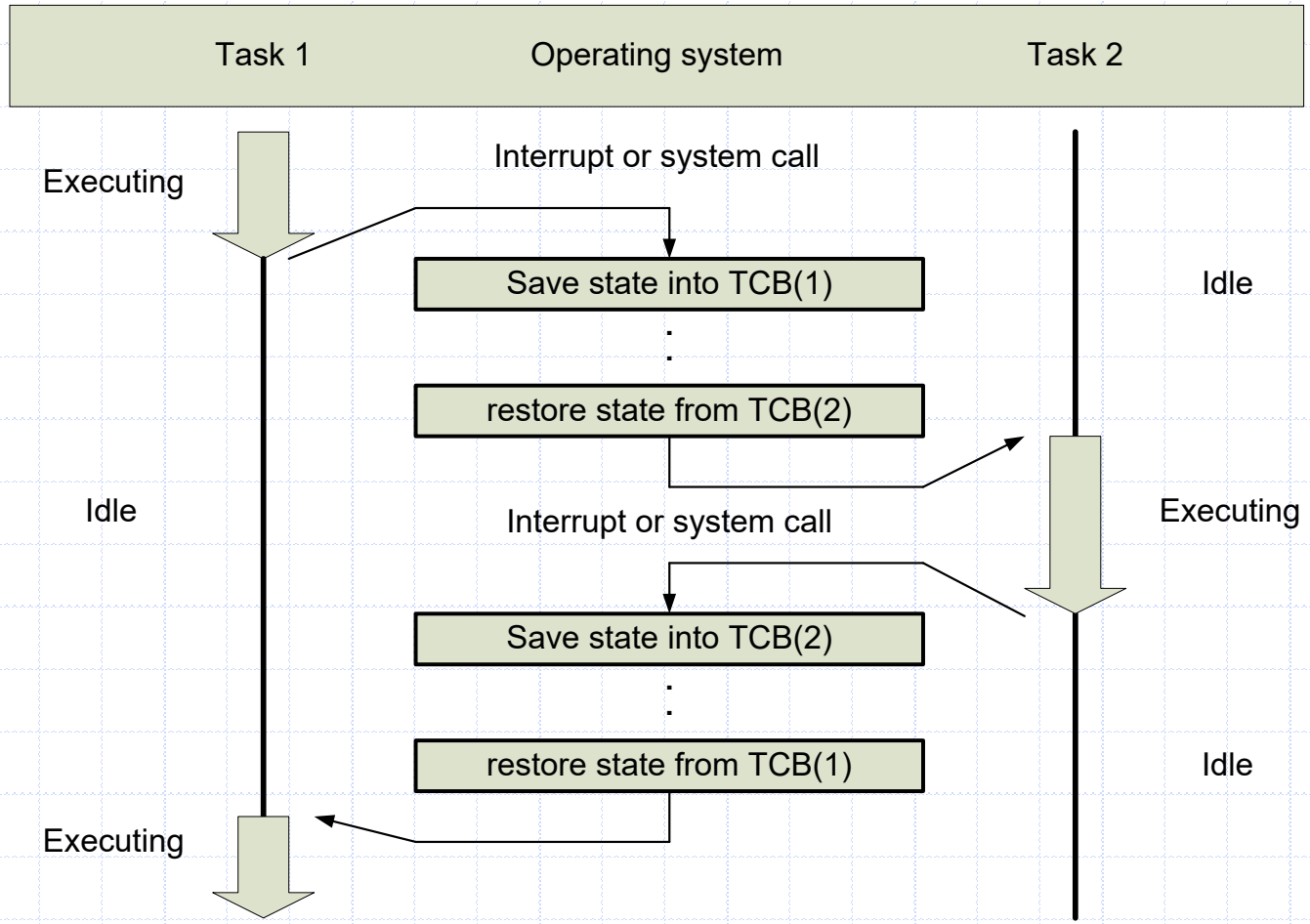
Multiprogramming between process A and process B

- Vi antager at der ligger to processer i process table
  - Process A
  - Pro
- Proc
  CPU'



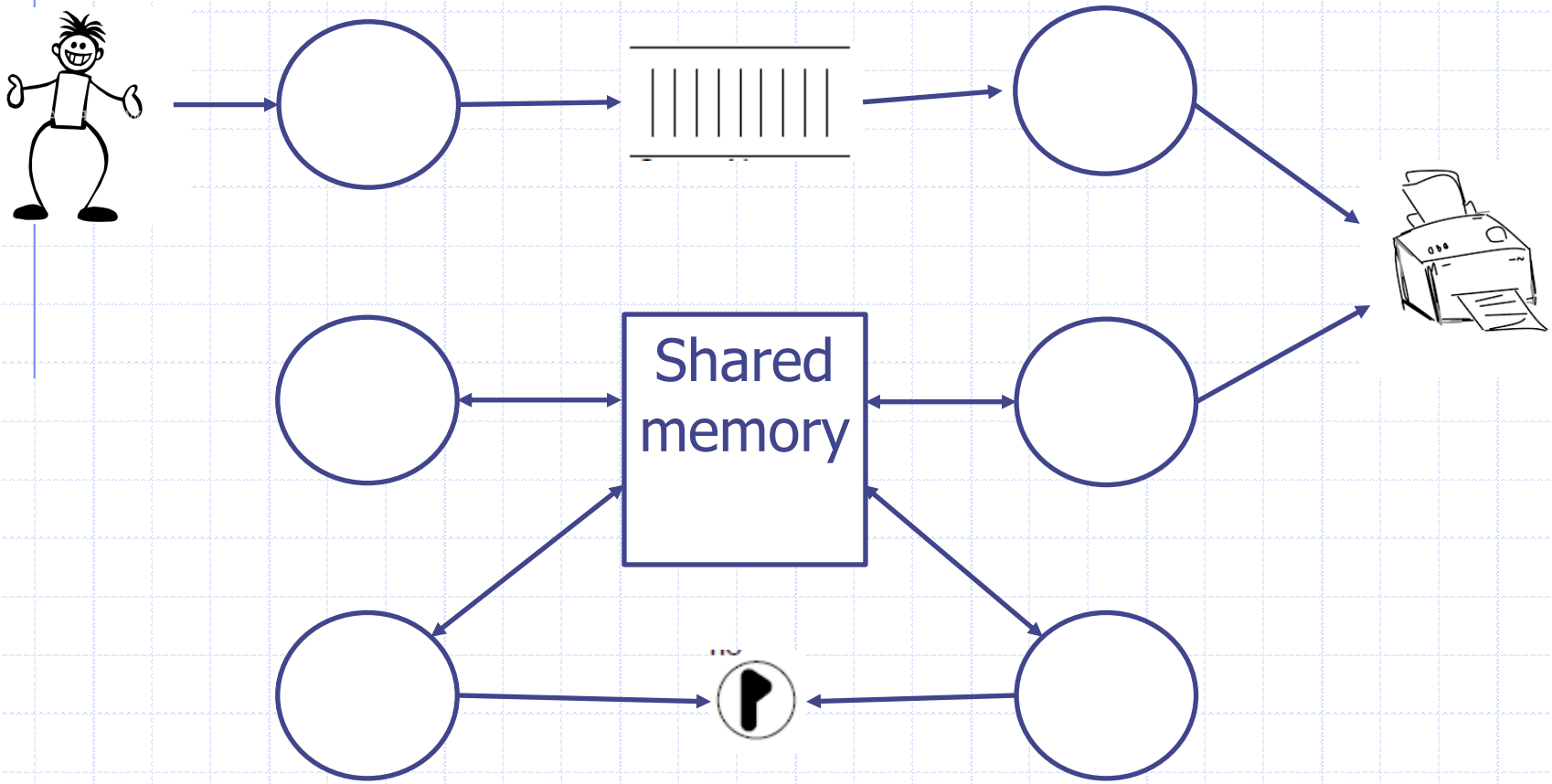
Morten Hansen
moh@sdu.dk

Context switch



Morten Hansen
moh@sdu.dk

Processdiagram

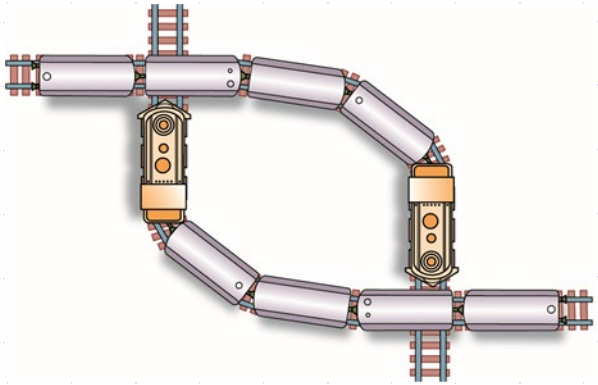


Morten Hansen
moh@sdu.dk

Håndtering af konkurrence blandt processer

- **Critical Region:** A group of instructions that should be executed by only one process at a time.
  - **Semaphore:** A “control flag”.
  - **Mutual exclusion:** Requirement that only one process at a time be allowed to execute a Critical Region.
- **Race condition:** Can happen in this region when more processes uses the same resource.
- **Deadlock**
  - Processes block each other from continuing because each is waiting for a resource that is allocated to another.
  - All these conditions are required for a deadlock:

1. Competition for non-sharable resources
2. Resources requested on a partial basis
3. An allocated resource can not be forcibly retrieved



Morten Hansen
moh@sdu.dk

Spørgsmål?

Morten Hansen
moh@sdu.dk