

Inleiding Operating Systems

Hoofdstuk 1



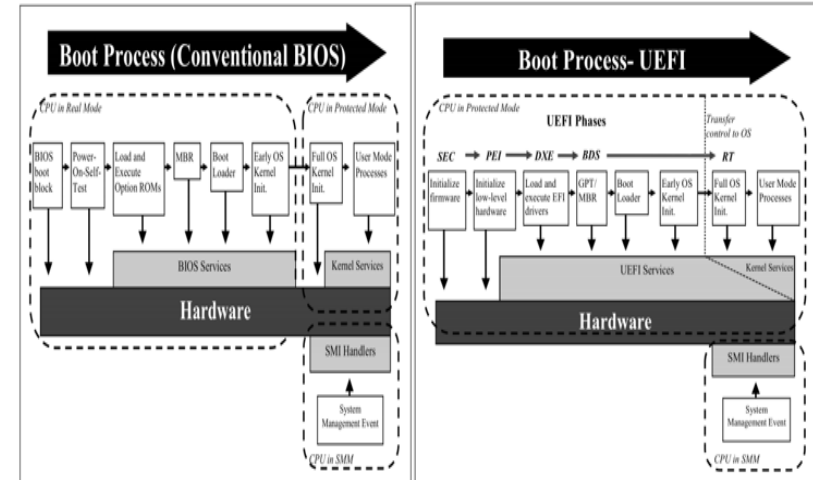
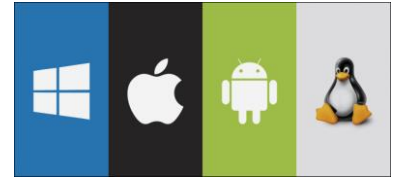
**DE HOGESCHOOL
MET HET NETWERK**

Elfde-Liniestraat 24, 3500 Hasselt, www.pxl.be



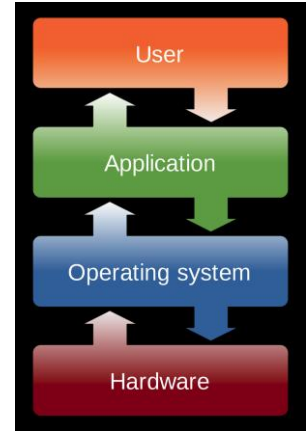
Operating Systems

- Computer aanzetten \Rightarrow niet meteen bruikbaar
 - Programma nodig dat hardware aanstuurt
 - Dit programma = **besturingssysteem (operating system of OS)**
- Onmiddellijk na het aanzetten van de PC start het boot process
- Verschil in boot mechanisme bij BIOS and UEFI systemen (later meer)
 - traditioneel BIOS :
beperkt in beveiliging
en aantal disks/partities om te managen
 - modernere UEFI
(Unified Extensible Firmware Interface) :
met beveiliging en tot 128 partities



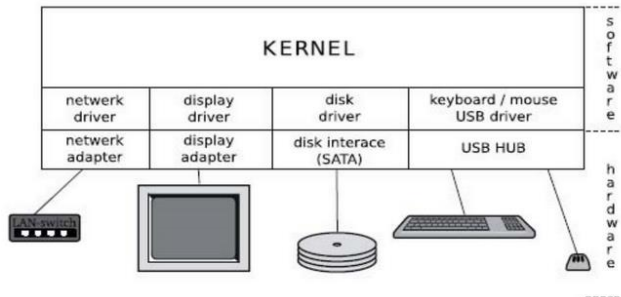
Taken van Operating System

- 1. Hardware toegang
 - Regelt **communicatie** met randapparaten (harde schijf, netwerk, USB,...)
 - Verdeelt, regelt en organiseert de **systemresources** (CPU-tijd, geheugen, diskruimte,...)
 - Je kan als gebruiker het systeem hardware **onafhankelijk** gebruiken bijv voor elk soort computer is Windows 11 altijd hetzelfde.
- 2. Bestands- en mappenbeheer
- 3. Gebruikersinterface
- 4. Applicatiebeheer
- 5. Ander functionaliteiten



Kernel, device drivers

- **Kernel:** (De kern van besturingssysteem)
 - Software.
 - **Hardwareonafhankelijk.**
- **Device drivers:**
 - Software.
 - Laat computer communiceren met randapparaten of devices.
 - Kan je toevoegen aan OS. Na installatie mogelijks pc opnieuw opstarten.
 - **Hardwareafhankelijk** (want apparaten hebben verschillende hardware)



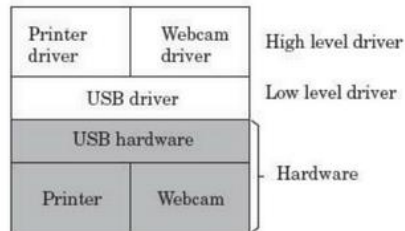
Device drivers

- **Low-level device driver:**

- Doet de eigenlijke aansturing van de hardware.
- Zit in kernel space.

- **High-level device driver:**

- Specifiekere toepassing van hardware.
- Gebruikt de low-level driver.
- Zit in user space.



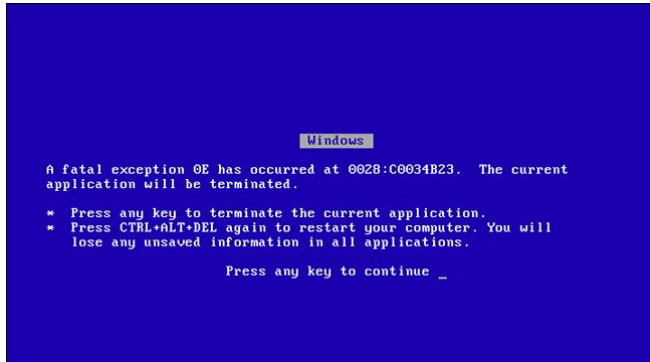
Wat is kernel space of user space?

User space vs kernel space

- **Kernel space:**
 - Hierin draait de software van het besturingssysteem
- **User space:**
 - Hierin draait de Software van applicaties (programma's) die op besturingssysteem lopen.

⇒ Gescheiden!

- Vroeger: als applicatie crashte, crashte hele besturingssysteem ook.



Process

- **Process:**

- Applicatie of **programma dat uitgevoerd wordt.**
- Ontstaat bij starten van programma en verdwijnt als het klaar is.
- (demo)
 - Start teksteditor op, typ C broncode in terwijl de teksteditor aan het uitvoeren is en sluit het venster.
 - Start compiler op, de code compileert de broncode naar machinetaal en de compiler sluit zich af.
 - Start de nieuwe gecompileerde applicatie op, deze voert uit totdat de gebruiker een actie doet om te sluiten of sluit automatisch na bepaalde handelingen.

Process is ooit broncode geweest in C, C++, C#, Java, Python,...

Even tussendoor... (niet vanbuiten leren)

- **Broncode** geschreven door een programmeur = tekst

vb. in bestand programma.c

```
#include <stdio.h>
```

```
int maal_drie(int a) {  
    return a * 3;  
}
```

```
int main (void) {  
    int x = maal_drie(3);  
    printf("Uitkomst is %d", x);  
    return 0;  
}
```

- **Compiler** compileert de broncode via een commando:
- **Naar een uitvoerbaar programma met machinecode:**

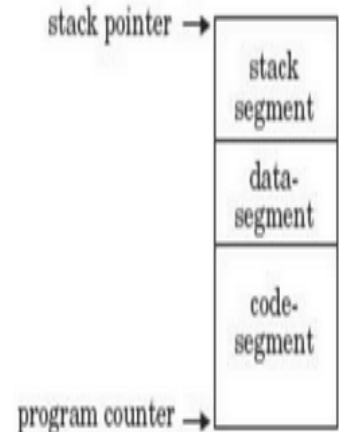
vb. gcc -o programma.c

vb. programma.exe

<https://www.youtube.com/watch?v=WiaiZmWxcck>

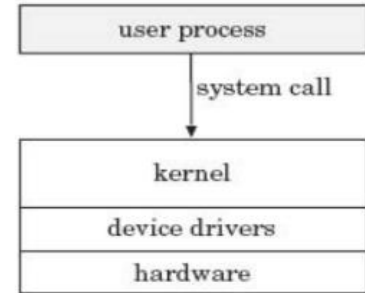
Process in computergeheugen

- **Process** in computergeheugen: na opstarten van gecompileerd programma
 - **Codesegment:**
 - Machine-instructies
 - **Program counter:** heeft adres van instructie die wordt uitgevoerd door CPU
 - Enkel gelezen door CPU. NIET geschreven (code mag zichzelf niet aanpassen!)
 - **Datasegment:**
 - Data of gegevens waarmee de machine-instructies gaan rekenen
 - Gelezen en geschreven door CPU.
 - **Stacksegment:**
 - Tijdelijke gestapelde gegevens waar **stack pointer** naar wijst.
 - vb/ returnadressen van functies
 - Gelezen en geschreven door CPU.



System calls

- Process kan **system call** aanvragen aan OS.
- **System call = speciale kernel routine van OS oproepen om bepaalde taak te doen.**
 - vb/ Bestand op harde schijf lezen.
- System calls zitten in een bibliotheek van OS.
- Machine-instructies zitten in de processor.
 - Beiden worden gebruikt door device drivers.



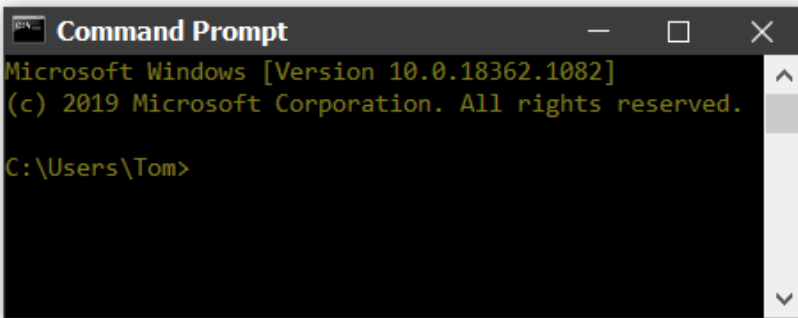
Voorbeelden van system calls op een Unix OS (niet studeren)

- create: nieuwe file maken
- open: file openen
- close: file sluiten
- pipe: leg communicatiekanaal tussen 2 processen
 en stuur data door in volgorde van verzenden (FIFO)
- fork: nieuw process maken
- kill: signaal naar een (ander) process sturen

Command interpreter (Windows) of shell (Unix)

- Speciaal programma samen met besturingssysteem geleverd.
- Je geeft commando's in met toetsenbord.
- De command interpreter "interpreteert" commando's (via system calls) + voert ze uit als process.
- We noemen dit de **CLI of Command Line Interface**
- Je kon in DOS pas nieuwe commando's typen als vorige commando afgelopen was.
 - Vaak draaien er tegenwoordig achtergrondprocessen tegelijk met de command interpreter

(demo) → Windows-toets + R ⇒ cmd of powershell



Soorten besturingssystemen

INDELING 1

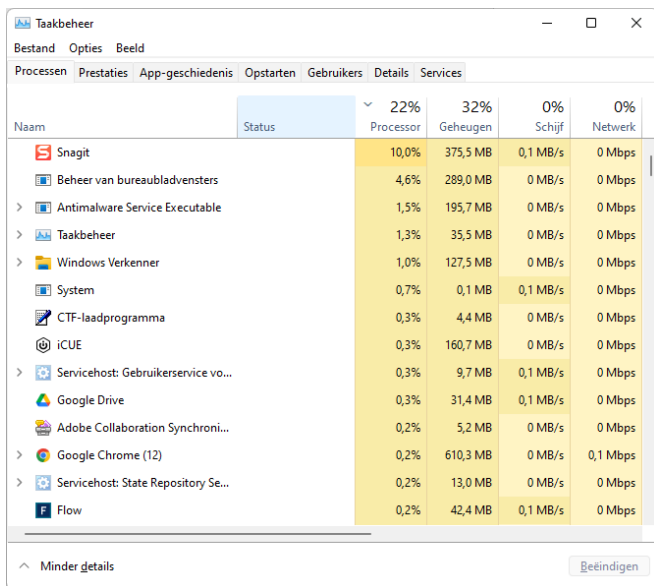
- **Single-tasking operating system:**
 - kan 1 process om beurt uitvoeren
- **Multi-tasking operating system:**
 - kan meer processen schijnbaar tegelijk
 - **Single-user:** 1 gebruiker
 - **Multi-user:** meerdere gebruikers

INDELING 2

- **Realtime:** OS moet binnen bepaalde tijd op een gebeurtenis reageren
vb/ vliegtuigcode, medische apparatuur
- **Niet-realtime:** tijd speelt geen rol

Multi-tasking

- Meerdere processen zijn **schijnbaar** tegelijk actief.
 - Echt: 1 process tegelijk actief per CPU (indien geen dual core, quad core, ...) en CPU wisselt vaak van process.
 - Lijkt alsof alle processen parallel draaien.



The screenshot shows the Windows Task Manager window titled 'Taakbeheer'. The 'Processen' tab is selected, displaying a list of running processes. The columns shown are 'Naam', 'Status', 'Processor', 'Geheugen', 'Schijf', and 'Netwerk'. The processes are sorted by CPU usage, with 'Snagit' at the top (10.0%) and 'Flow' at the bottom (0.2%).

Naam	Status	22% Processor	32% Geheugen	0% Schijf	0% Netwerk
Snagit		10,0%	375,5 MB	0,1 MB/s	0 Mbps
Beheer van bureaubladvensters		4,6%	289,0 MB	0 MB/s	0 Mbps
> Antimalware Service Executable		1,5%	195,7 MB	0 MB/s	0 Mbps
> Taakbeheer		1,3%	35,5 MB	0 MB/s	0 Mbps
> Windows Verkenner		1,0%	127,5 MB	0 MB/s	0 Mbps
System		0,7%	0,1 MB	0,1 MB/s	0 Mbps
CTF-laadprogramma		0,3%	4,4 MB	0 MB/s	0 Mbps
iCUE		0,3%	160,7 MB	0 MB/s	0 Mbps
> Servicehost: Gebruikersservice vo...		0,3%	9,7 MB	0,1 MB/s	0 Mbps
Google Drive		0,3%	31,4 MB	0,1 MB/s	0 Mbps
Adobe Collaboration Synchroni...		0,2%	5,2 MB	0 MB/s	0 Mbps
> Google Chrome (12)		0,2%	610,3 MB	0 MB/s	0,1 Mbps
> Servicehost: State Repository Se...		0,2%	13,0 MB	0 MB/s	0 Mbps
Flow		0,2%	42,4 MB	0,1 MB/s	0 Mbps

At the bottom of the window, there is a 'Minder details' link and a 'Beëindigen' button.

Multi-tasking

- **Process table:**

- Tabel in de kernel met processen en hun toestanden.

- **Running:**

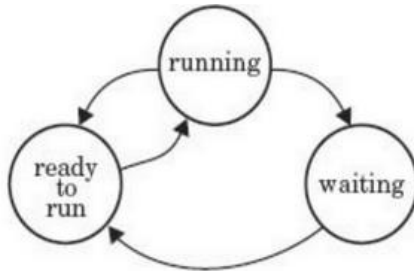
- process wordt uitgevoerd (1 per CPU-core)

- **Ready to run:**

- process is klaar om gekozen te worden om te runnen

- **Waiting:**

- process wacht op een **event** om opnieuw ready te zijn
- bij event treedt interrupt op vb/ drukken op keyboardtoets



Process ID	State	Privileged	Process Name	Priority	Register Set	File List
45	Ready	No	iexplorer.exe	Medium	PC=1000, AX=3421	empty
50	Running	No	word.exe	Medium	PC=4000, AX=1111	Mylecture.doc
81	Blocked (waiting)	No	Outlook.exe	Medium	PC=8321, AX=1	empty
720	Ready	Yes	winlogon.exe	High	PC=34, AX=-1234	empty

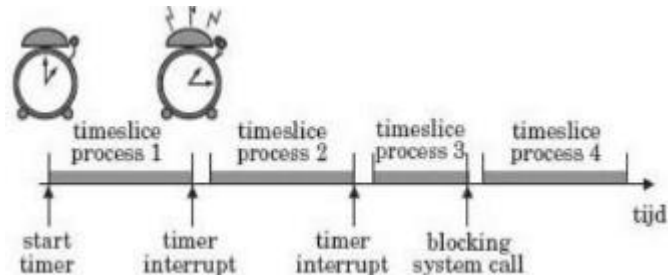
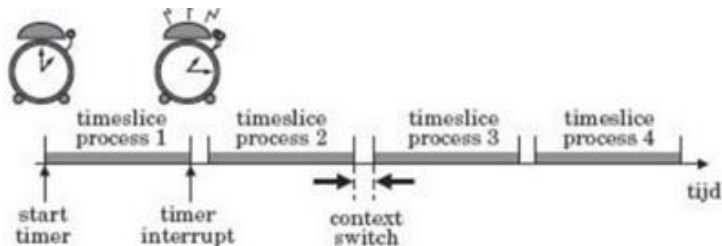
Scheduling

- **Scheduler** bepaalt welke processen door besturingssysteem worden uitgevoerd.
 - Kiest ready to run process uit process table.
 - Geeft dit aan CPU.
 - Pre-emptive of non-pre-emptive. (Pre-emptive = onderbrekend)



Pre-emptive scheduler

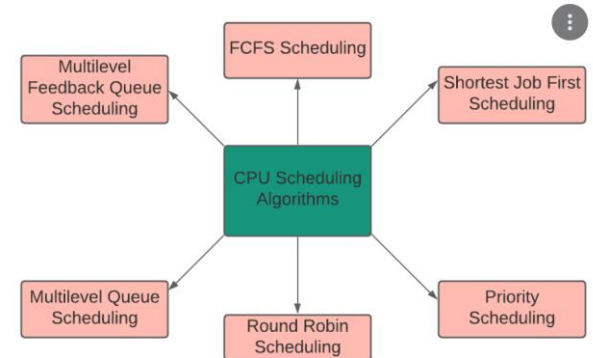
- Scheduler **onderbreekt** process.
- Runt ander process (met hogere prioriteit).
- Elk process krijgt tijd om op CPU te draaien = **timeslice**
- Timer genereert **interrupt**.
- **Context switch** (wisselen van process).
- Onderbroken process kan later hervat worden.
- **Blocking system call:**
 - onderbreek process vroeger dan timeslice \Rightarrow **waiting**



Pre-emptive scheduler

- Hoe wordt process gekozen bij context switch?
 - **Round robin:**
 - Alle processen in een wachtrij.
 - Process aan de beurt geweest
 - Sluit vanachter terug aan om verder te gaan.
 - **Prioriteiten:**
 - Processen met hoge prioriteit (belangrijker) gaan voor!
 - **Combinatie:**
 - 2 processen met dezelfde **prioriteit**,
→ terugvallen op **round robin**.
- **Slack:**
 - Tijd die een afgebroken process nog nodig heeft om klaar te geraken.

Round Robin is a CPU scheduling algorithm where each process is cyclically assigned a **fixed time slot**. It is the preemptive version of First come First Serve CPU Scheduling algorithm. Round Robin CPU Algorithm generally focuses on **Time Sharing** technique.



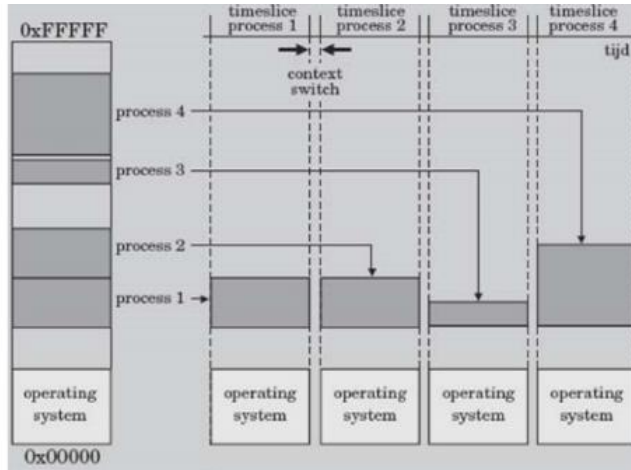
Non-pre-emptive scheduler

- Scheduler laat process runnen totdat het afgelopen is en onderbreekt het dus niet.
- Daarna pas next-process system call + context switch.
- Komt minder vaak voor, want prioriteit is belangrijk!



Memory management

- Bij elke CPU zit in hardware meestal een **MMU (memory management unit)**
 - Vertaalt CPU-adres naar werkgeheugen-adres waar echte process zit.

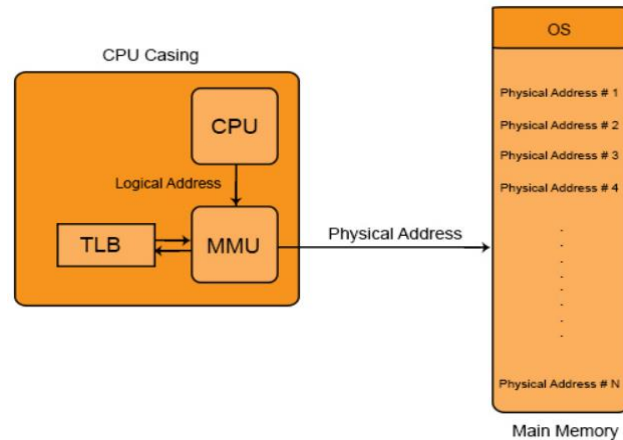


(pijlen verwarrend getekend...)

- Eenvoudigste geval: Vast getal optellen of aftrekken bij CPU adres
- Meestal complexer: Gedeelten van process verspreid over geheugen

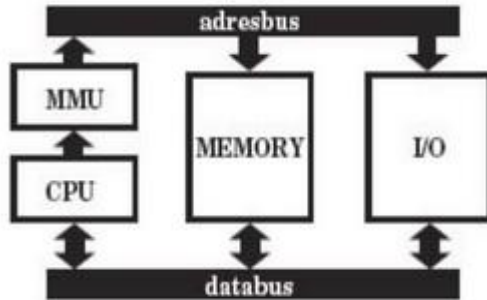
Werking van MMU

- Nieuw process start op.
- CPU zoekt vrij geheugen.
- CPU vertelt de MMU waar het process écht is.
- **MMU** vertaalt **CPU-adressen** dan naar **echte adressen**.
 - Maar 1 process tegelijk!
- Lijkt alsof alle actieve processen op hetzelfde geheugenadres beginnen.
- Er is maar **1 process per CPU** echt tegelijk bezig
⇒ geen problemen



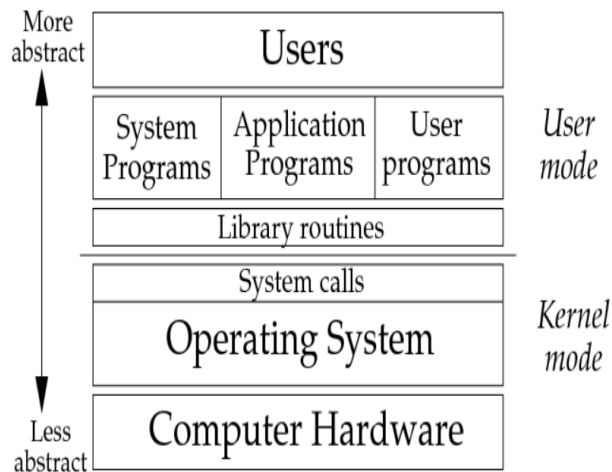
MMU

- Waar zit MMU?
 - Intern in de CPU zelf (moderne Intel).
 - Afzonderlijke chip of hardware schakeling.



User mode en kernel mode

- **User mode (= protected mode):**
 - User processen.
 - Bepaalde instructies mogen niet uitgevoerd worden, enkel via system calls.
 - Interrupt brengt CPU naar kernel mode (enkel via **system calls!**).
- **Kernel mode (= supervisory mode):**
 - System calls worden door kernel uitgevoerd.
 - Machine code van besturingssysteem uitvoeren.
 - CPU kan terugkeren naar user mode.



Processmanagement bij beperkt geheugen

- Stel: Processen geraken niet allemaal in vol werkgeheugen (**RAM**).
- Oplossingen (buiten meer RAM-chips kopen)
 - 1) Geen nieuwe processen meer toelaten totdat er genoeg processen klaar zijn.
 - 2) Beter: **swapping**
 - OS zoekt naar wachtend process.
 - Schrijf wachtend process even naar disk.
 - Gebruik vrijgekomen geheugen.
 - ⇒ **virtueel geheugen**
 - (heel process is 1 brok)
- 1) Best (tegenwoordig):
 - Ingewikkeldere MMU + CPU met **demand paging**
 - ⇒ **virtueel geheugen**

Demand paging

What is Demand Paging?

Demand paging, is a technique used in virtual memory systems where the **pages** are **brought** in the **main memory** only when **required or demanded by the CPU**.

- **Demand paging (Virtueel geheugen):**
 - Processen opdelen in pages: kleine stukjes van vaste grootte.
 - 512 of 4096 bytes
 - Pages enkel in **RAM** laden wanneer **nodig**.
 - Voorzie 1 vrije plaats op **disk** om page in te laten **wachten**.
 - Als CPU klaar is met page:
 - Verwittig de MMU.
 - MMU stuurt **page trap** signaal naar CPU.
 - CPU komt in kernel mode.
 - CPU haalt nieuwe page van process op van de disk
 - Zet die page in RAM.
 - Als RAM vol is: dan wordt de huidige page van een ander process op vrije plaats op disk gezet.
 - Herhaal tot process afloopt.

Demand paging

- **Voordelen van demand paging:**

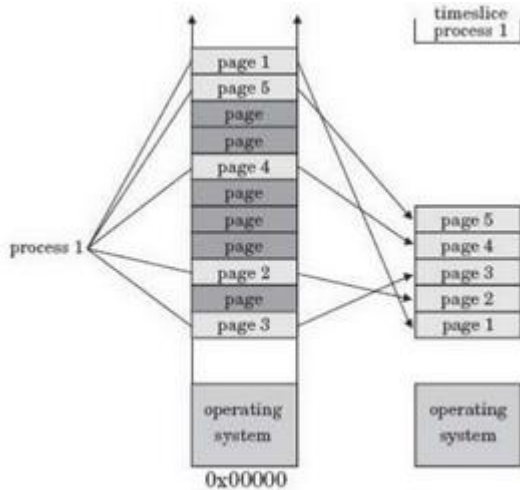
- 1) Sneller: **pages** wegschrijven naar disk **ipv. hele process**.
- 2) Sneller: pages die amper nodig zijn (error handling) **niet in RAM** geladen.
- 3) Je kan processen verwerken die **groter** zijn dan **beschikbare RAM**.
- 4) Geen probleem om passend aaneensluitend stuk geheugen te vinden voor nieuw process. Bij swapping wel probleem, want 1 geheel....

- **Nadelen van demand paging:**

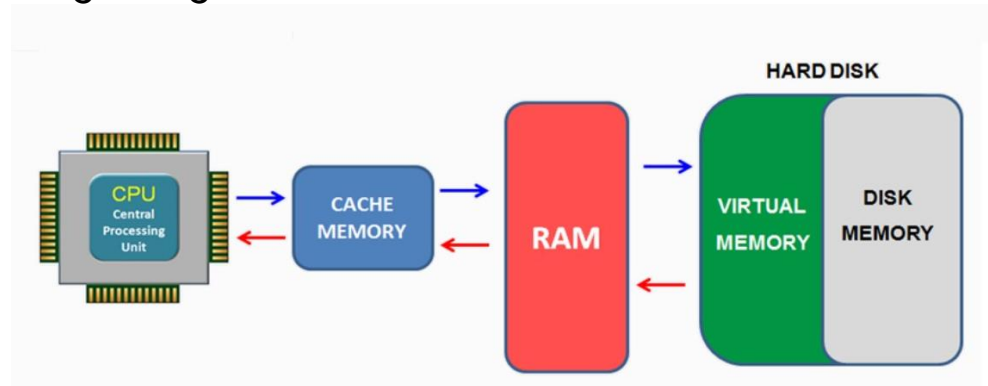
- 1) Alleen mogelijk als CPU hardware met paging omkan.
- 2) MMU moet hier ook op voorzien zijn.
- 3) Snelheid om process uit te voeren hangt af van onvoorspelbare externe factoren: ⇒ **niet realtime**
 - Aantal processen
 - Hoeveelheid geheugen

Virtueel geheugen

- **Virtueel geheugen:** computer krijgt schijnbaar groter werkgeheugen, groter dan fysiek geheugen.

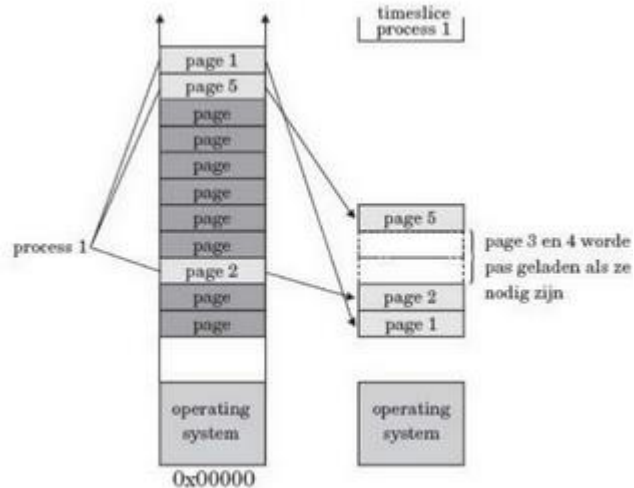


- Pages van process 1 door elkaar op disk.
- Process 1 wordt uitgevoerd voor bepaalde timeslice.
- MMU zet alle pages goede plaats in geheugen.



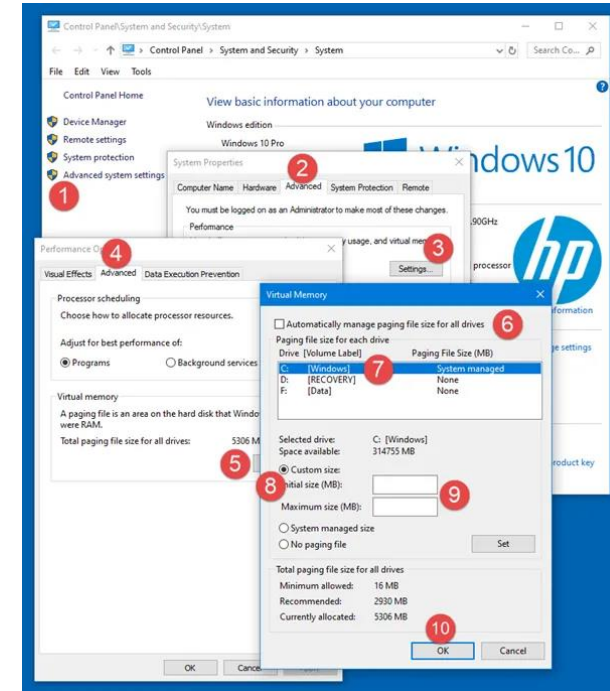
Virtueel geheugen

- Niet alle pages hoeven aanwezig te zijn.
- Ze worden pas geladen wanneer ze nodig zijn.



Virtueel geheugen

- **Swap space:**
 - Ruimte op harde schijf om pages naar te swappen.
- **Swap file:**
 - Andere besturingssystemen werken met een file (bestand).



Shared code

- **Code segment van process:**

- Machine instructies. ⇒ wat process doet!
- Enkel gelezen door CPU. NIET geschreven (code mag zichzelf niet aanpassen!)

⇒ Processen kunnen codesegmenten delen = **shared code!**

Dynamic link libraries (DLL's), shared libraries

- Compiler vertaalt C-code naar machinecode. (demo)
 - Executable: een programma vb/ .exe file
 - **DLL/shared library:**
 - Voorgecompileerde bibliotheek van routines die door een programma gebruikt kunnen worden (vb/ printf in C).
- **Linken:** machine code van bibliotheekroutines koppelen aan code van programma.
 - **Static linking:**
 - als process runt, wordt alle code in geheugen geladen, ook bibliotheek.
 - **Dynamic linking:**
 - linken pas doen als process de bibliotheek nodig heeft.
- **DLL's / shared libraries:** bibliotheek met routines die je dynamisch kan linken.

File management

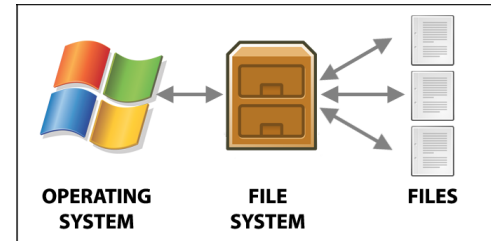
- Bestanden of files:
 - Gegevens, teksten, foto's, video's,...
 - Staan meestal op disk.
- **Filenaam:**
 - Mag enkel bepaalde tekens hebben (bepaald door filesystem).
 - Al dan niet onderscheid tussen grote of kleine letters.
- **Extensie:**
 - vb/ .exe op Windows voor een programma, .txt voor tekstfile,...

.jpg .mp4
.txt

Filesystem

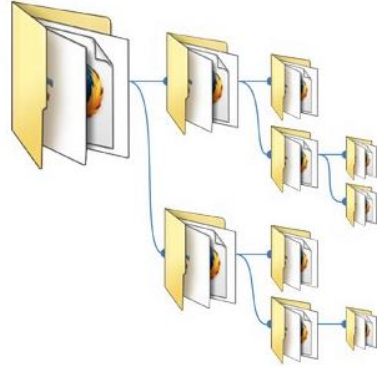
- **Filesystem:**

- De manier waarop de disk georganiseerd is en hoe we files kunnen bereiken.
- Bewerkingen: (demo)
 - Aanmaken van bestand.
 - Wijzigen van bestand.
 - Opvragen van de inhoud.
 - Verwijderen van bestand.



File management: directories

- Bestanden geordend volgens boomstructuur:
 - **Directory** (map):
 - Bestanden
 - **Subdirectories** (mappen in andere mappen):
 - Bestanden
 - Subdirectories
 - ...
- (demo)



Interrupts

- I/O-device stuurt een **interrupt** naar CPU zodat CPU iets kan doen.
 - Voorbeeld: toetsenbord
- Laat CPU pas op I/O reageren als het nodig is.
- Laat CPU **géén** tijd verspillen met **polling**.
- **Werking van interrupt:**
 - I/O device zendt interrupt signaal uit
 - CPU reageert op interrupt signaal
- **Polling**
 - CPU op regelmatige basis laten vragen aan I/O-devices of er nog iets te doen is.
 - CPU reageert indien I/O-device iets te doen heeft...
 - Polling is niet zo efficiënt.
 - Interrupts zijn beter.

Realtime besturingssysteem

- **Realtime besturingssysteem**

- Heeft realtime scheduler.

⇒ je wil dat sommige taken binnen een bepaalde tijd moeten gebeuren
vb/ in auto's, medische apparatuur, vliegtuigcode,...

Job en task

- **Job:** een klus die geklaard moet worden.
 - Heeft een starttijd (**release time**).
 - Er moet op een zeker moment aan job begonnen worden.
 - Heeft een **deadline in een realtime systeem**.
 - Tijdstip waarop job klaar MOET zijn. (hangt af van toepassing)
- **Task:** groter geheel waar bepaalde job kan bijhoren
- **Sporadic job:**
 - Job die naast de standaard jobs af en toe opduikt.
 - vb/ reactie van besturingssysteem op ingreep van systeembeheerder
- **Periodic job:**
 - Job die om de zoveel tijd steeds herhaald moet worden.
 - Vaak in realtime systemen.

Timing

- **Hard realtime systeem:**
 - Systeem met een deadline die gehaald moet worden.
 - Voorbeeld: industriële processen, medische apparatuur, automatische piloot.
- **Soft realtime systeem:**
 - Deadline mag binnen zekere marge overschreden worden.
 - Kan een volgende keer gecompenseerd worden.
 - Voorbeeld: communicatiesysteem met “**Quality of Service**” (QoS)
 - Garantie geven dat je gegevens met zekere snelheid kan versturen.
⇒ soft realtime
 - Gemiddeld gezien. Dus soms wordt deadline overschreden.

Static vs. dynamic scheduling

- **Static scheduling:**
 - Scheduling ligt volledig vast: jobs gebeuren in **vaste volgorde**.
 - Simpeler, maar minder flexibel.
- **Dynamic scheduling:**
 - Scheduler bepaalt volgorde van jobs.
 - **Prioriteiten en deadlines** bepalen welke job op CPU komt.
 - Flexibeler:
 - kan in verschillende situaties gebruikt worden.

Offline vs. online scheduling

- **Offline scheduling:**

- Scheduling doen voordat het systeem gestart wordt.
- Voorbeeld:
 - Op voorhand invoeren van bewegingspatronen in een industriële robot.



- **Online scheduling:**

- Terwijl systeem runt.

Scheduling in standaard besturingssysteem

- **Window of Unix/Linux:**
 - **Pre-emptive dynamic scheduler (ook online)**
 - op basis van prioriteitsklassen.
 - Round robin.
 - Jobs krijgen timeslices.
 - **NIET REALTIME!, want:**
 - Virtueel geheugen.
 - Interrupts.
 - Kernel taken krijgen vaak voorrang.
 - Deze eigenschappen verbeteren performance, maar minder voorspelbaar.
 - Worst-case zit job bijvoorbeeld niet in RAM, maar op disk geswapt.

Haalbaarheidsanalyse

- **Wat als scheduler in realtime systeem niet haalbaar is?**
 - Kies snellere CPU.
 - Los bottlenecks op.
 - Verschuif deadlines. Misschien té strak? \Rightarrow **soft realtime**
 - Multiprocessorsysteem gebruiken of iets paralleler.