



H1 – Besturingssystemen

Inleiding

**HO
GENT**

1. Inleiding Besturingssystemen

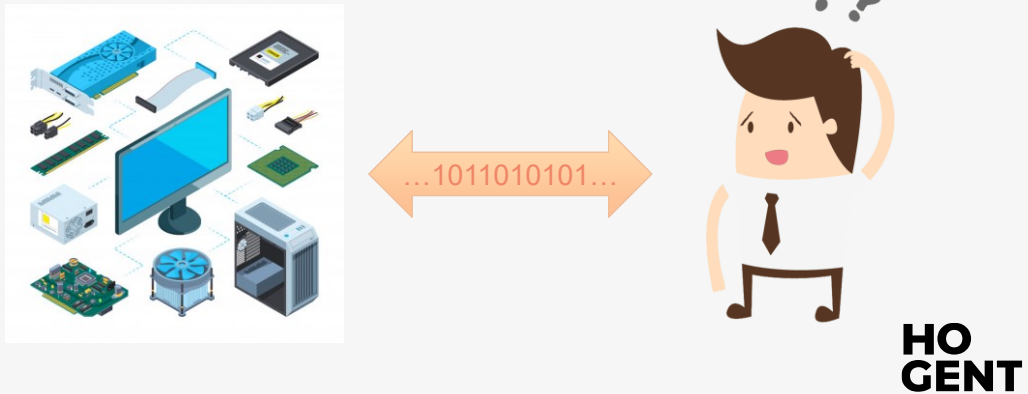
- 1.1 Wat is een besturingssysteem?
- 1.2 Soorten besturingssystemen
- 1.3 Concepten van besturingssystemen

**HO
GENT**

1.1 Wat is een besturingssysteem?

**HO
GENT**

Link tussen hardware en gebruiker



Link tussen hardware en gebruiker

Zonder een besturingssysteem zou het bijna onmogelijk zijn om een computer te gebruiken. Als gebruiker zou je rechtstreeks de hardware moeten aanspreken, en zou je er ook zelf voor moeten zorgen dat de hardware kan samenwerken. Een eenvoudige taak, zoals een tekstbestand openen en de inhoud op het scherm tonen of doorsturen naar een printer, zou al enorm complex worden..

Wat is een besturingssysteem?



- **Programma** dat het mogelijk maakt de hardware van een computer te gebruiken
- Hardware = CPU, geheugen, secundaire opslagapparatuur, randapparatuur, ...
- Gebruikers geven geen opdrachten aan de computer/hardware, maar aan het besturingssysteem
- Het besturingssysteem geeft de hardware de opdracht om de gewenste taken uit te voeren
- EN: Operating System (OS)

**HO
GENT**

Wat is een besturingssysteem?

In hoofdzaak is een besturingssysteem dus een programma dat mensen de mogelijkheid geeft gebruik te maken van de hardware van een computer (CPU's, geheugen en secundaire opslagapparatuur).

Gebruikers geven geen instructies aan de computer, maar in plaats daarvan aan het besturingssysteem. Het besturingssysteem geeft de hardware de opdracht de gewenste taken uit te voeren.

De Engelse term voor besturingssysteem is "Operating System", de afkorting OS wordt vaak gebruikt.

Taken van een besturingssysteem



- Opslaan en ophalen van informatie
- Programma's afschermen
- Gegevensstroom regelen
- Prioriteiten regelen
- Beheer en delen van bronnen
- Tijdelijke samenwerking tussen programma's mogelijk maken
- Reageren op fouten
- Tijdsplanning maken
- ...

**HO
GENT**

Taken van een besturingssysteem

Veel mensen willen van systeembronnen (resources) gebruik maken. Als er geen besturingssysteem zou bestaan, zouden er conflicten optreden en zou er een enorme verwarring ontstaan. Een besturingssysteem moet zoveel mogelijk gebruikers gelukkig maken.

Enkele taken van een besturingssysteem:

- programma's de mogelijkheid geven informatie op te slaan en terug te halen;
- programma's afschermen van specifieke hardware zaken;
- de gegevensstroom door de componenten van de computer regelen;
- programma's in staat stellen te werken zonder door andere programma's te worden onderbroken;
- onafhankelijke programma's de gelegenheid geven tijdelijk samen te werken en informatie gemeenschappelijk te gebruiken;
- reageren op fouten of aanvragen van de gebruiker;
- een tijdsplanning maken voor programma's die resources willen gebruiken.

Over het algemeen kunnen we zeggen dat een besturingssysteem zoveel mogelijk

gebruikers gelukkig moet maken. Natuurlijk is het geluksgevoel van de gebruiker afhankelijk van zijn verwachtingen.

Enkele voorbeelden



Enkele voorbeelden



Traditioneel denken vele gebruikers dat Windows het enige besturingssysteem is, of dat er slechts één alternatief is genaamd MacOS. Er zijn echter veel meer besturingssystemen.

Onder IT'ers is het Linux besturingssysteem wijd verspreid. De parallellen met één van de oudste besturingssystemen, Unix, zijn groot. O.a. het bedrijf Oracle verkoopt nog steeds Unix.

Vergeet niet dat er ook heel wat besturingssystemen bestaan hebben, die nu niet meer bijgewerkt worden. Rekenmachines, klassieke GSM toestellen, ... hebben ook allemaal hun eigen besturingssysteem, ontworpen voor de specifieke hardware van die toestellen.

1.1 Wat is een besturingssysteem?

Enkele voorbeelden

	Microsoft 	Apple 	Andere 
x86/ x64 laptop/desktop	Windows 10 Windows Server 2019	MacOS 11	Fedora 33 Ubuntu 20.04.2 LTS CentOS 8 ...
Mobile devices	Windows 10 Mobile	iOS 14 iPadOS 14	Android 11 Plasma Mobile Ubuntu Touch ...
ARM devices	Windows 10 ARM	MacOS 11 (for M1 chips)	Raspberry Pi OS Fedora 33 ARM Ubuntu 20 Server ARM CentOS 8 ARM ...

**HO
GENT**

1.2 Soorten besturingssystemen

**HO
GENT**

Soorten besturingssystemen

We kunnen een onderscheid maken tussen:

- Single-tasking systemen
- Multitasking, single-user systemen
- Multi-user systemen

Single-tasking: er kan maar **1 taak** gelijktijdig actief zijn

Multi-tasking: **meerdere taken** kunnen gelijktijdig actief zijn

Multi-user: **meerdere gebruikers** kunnen gelijktijdig het systeem gebruiken

**HO
GENT**

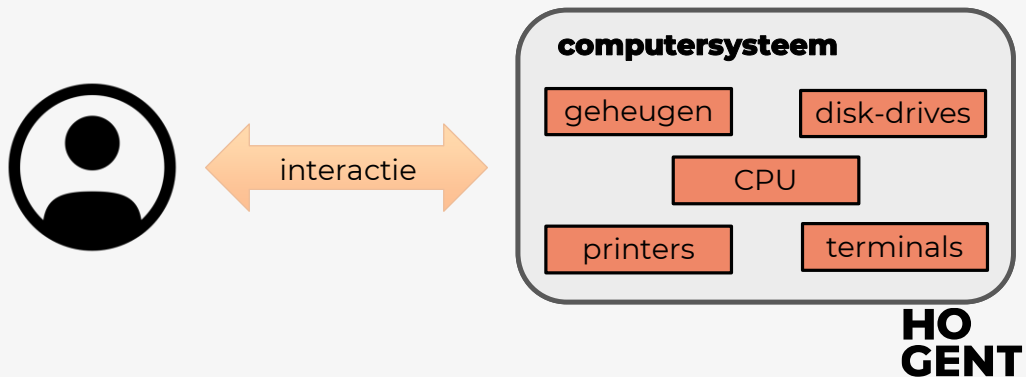
Soorten besturingssystemen

Omdat er veel verschillende soorten computers bestaan, zijn er ook veel verschillende soorten besturingssystemen. Sommige eenvoudige besturingssystemen stellen alle resources in dienst van één applicatie tegelijk. Andere geven de gebruiker de mogelijkheid verscheidene applicaties gelijktijdig te laten uitvoeren. Nog ingewikkelder systemen kunnen tegelijk de behoeften van vele gebruikers aan.

Een besturingssysteem moet echter op meer dingen letten dan alleen het aantal programma's of gebruikers. In sommige gevallen is het voldoende om programma's na elkaar te verwerken. Er is dan geen noodzaak aanwezig om snel te reageren. Aan de andere kant proberen sommige besturingssystemen er voor te zorgen dat snel aan de verlangens van de gebruiker wordt voldaan. In weer andere gevallen moet een systeem binnen een bepaald tijdsbestek reageren om rampzalige toestanden te voorkomen. Alles hangt af van de behoeften van de gebruikers, van wat ze verwachten en van de applicaties die ze draaien.

Single-tasking

= Systeem waarbij 1 gebruiker 1 applicatie tegelijk draait



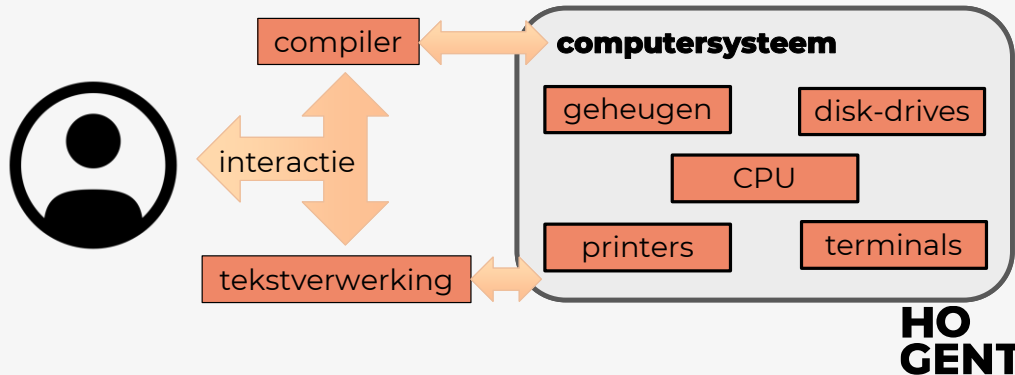
Single-tasking

Bij een single-tasking besturingssysteem is er maar één gebruiker die maar één applicatie tegelijk draait. In een dergelijk systeem kan er dus slechts één programma (taak/task) tegelijk actief zijn.

Een bekend voorbeeld van een single-tasking besturingssysteem is MS-DOS, de (niet-grafische) voorloper van Windows.

Multitasking (single-user)

= Meestal 1 gebruiker, die verscheidene taken tegelijkertijd kan uitvoeren



Multitasking (single-user)

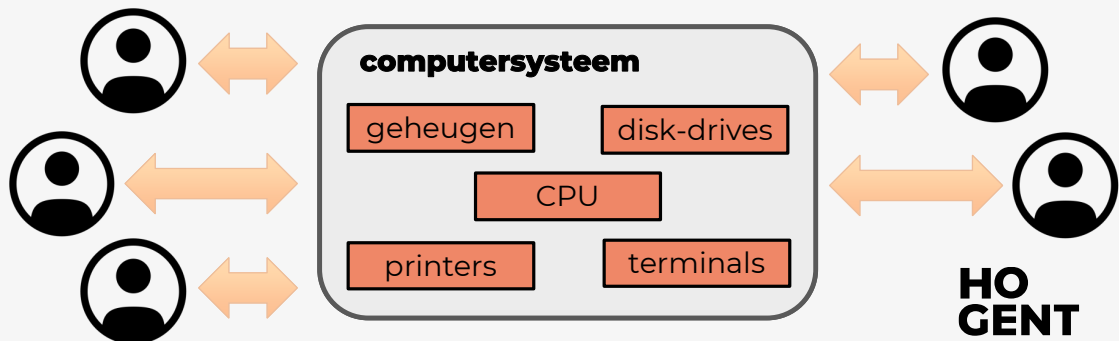
De meeste moderne besturingssystemen zijn multitasking, met andere woorden: meerdere programma's (taken) kunnen tegelijkertijd geopend en uitgevoerd worden.

Veel multitasking systemen staan nog steeds maar één gebruiker toe, maar hij/zij kan verscheidene bezigheden op hetzelfde moment laten afwikkelen.

Aangezien bij multitasking een gebruiker verschillende taken tegelijkertijd kan uitvoeren, worden bepaalde functies van het besturingssysteem (zoals bv. geheugenbeheer) ingewikkelder.

Multi-user systemen

= Meerdere gebruikers maken simultaan gebruik van de computerbronnen



Multi-user systemen

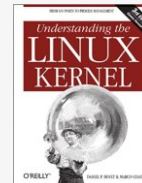
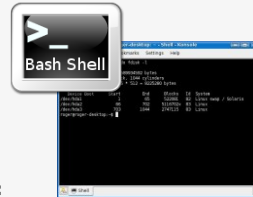
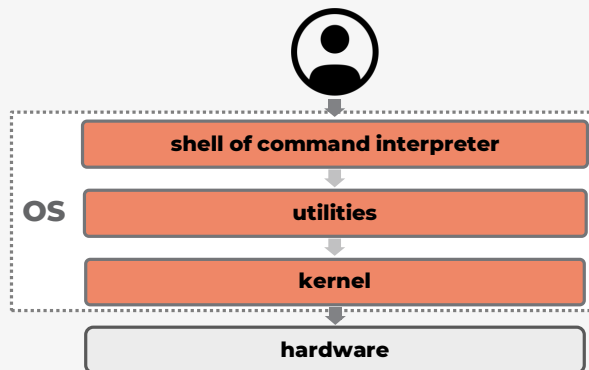
Multi-user systemen, ook wel multiprogrammering systemen genoemd, moeten niet alleen alle gebruikers bijhouden, er moet ook voorkomen worden dat deze elkaar hinderen of in het werk van de anderen kunnen rondneuzen.

Hierdoor worden bepaalde functies van een besturingssysteem nog complexer, want er kunnen meerdere gebruikers tegelijkertijd actief zijn op het systeem, en elke gebruiker kan ook meerdere taken tegelijkertijd uitvoeren.

1.3 Concepten van besturingssystemen

**HO
GENT**

Verschillende lagen



**HO
GENT**

Verschillende lagen

Veel besturingssystemen implementeren de interface tussen gebruiker en computer als een reeks stappen of lagen. In de toplaag zijn de functies vastgelegd en de onderste laag bevat de details van het laagste niveau om deze functies uit te voeren. De gebruiker communiceert met de bovenste laag. Deze bestaat uit routines van het besturingssysteem die zijn ontworpen om op opdrachten van een gebruiker te reageren. We noemen deze laag de shell of de command interpreter. Het is het deel van het besturingssysteem waarmee de gebruiker het meest vertrouwd is.

In werkelijkheid is het echter niet de shell die de opdracht van de gebruiker uitvoert. De reden daarvoor is dat veel opdrachten eigenlijk erg ingewikkeld zijn. Zo is het mogelijk dat er voor een eenvoudige opdracht die het besturingssysteem vraagt veel moet gebeuren.

De laag utilities bevat vele routines die voor deze dingen zorg dragen.

De laatste laag is de kernel of kern. Dit is het hart van het besturingssysteem. Deze laag bevat de routines die het vaakst worden gebruikt en waar het meest op aan

komt. Wanneer een gebruiker een opdracht geeft, verzorgen de utilities het grootste deel van de controle en de voorbereiding die voor de uitvoering nodig zijn.

Programma's / taken

Een besturingssysteem zorgt er voor dat taken (programma's) uitgevoerd worden. We kunnen een onderscheid maken tussen:

- Interactieve programma's (snelle respons)
- Batch programma's (geen directe respons)
- Real-time programma's (respons in een heel beperkte tijd)

**HO
GENT**

Soorten taken

Er bestaan verscheidene soorten multi-user-computers, afhankelijk van de soorten programma's die ze aankunnen:

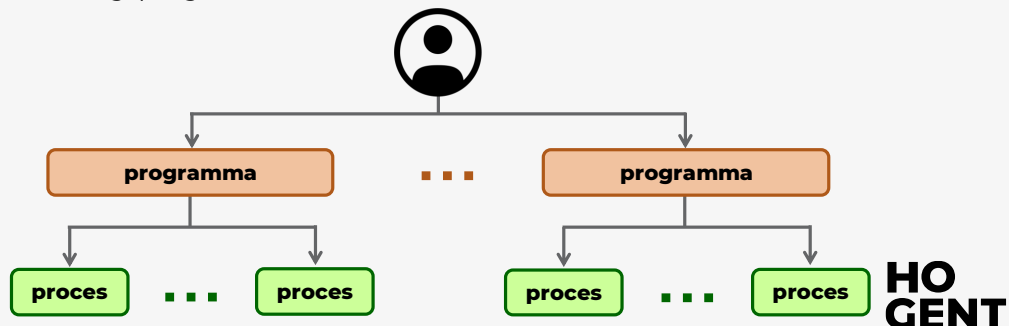
- **Interactieve programma's:** Een interactief programma is een programma dat een gebruiker vanaf de terminal activeert. Over het algemeen voert de gebruiker een korte opdracht in. Het besturingssysteem vertaalt deze opdracht en onderneemt actie. Het zet vervolgens een prompt-teken op het scherm en geeft daarmee aan dat de gebruiker een volgende opdracht kan invoeren. De gebruiker voert weer een opdracht in en het proces gaat door. De gebruiker werkt met het besturingssysteem op een conversatie-achtige manier, interactieve mode genoemd. Interactieve gebruikers verwachten een snelle respons. Daarom moet het besturingssysteem interactieve gebruikers voorrang geven.
- **Batch-programma's:** Een gebruiker kan opdrachten in een file opslaan en deze aan de batch queue (wachtrij voor batch-programma's) van het besturingssysteem aanbieden. Uiteindelijk zal het besturingssysteem de opdrachten uitvoeren. Batch-gebruikers verschillen van interactieve gebruikers omdat zij geen directe respons verwachten. Bij scheduling houdt het besturingssysteem hiermee rekening.
- **Real-time programma's:** Real-time programmering legt aan de respons een

tijdsbeperking op. Het wordt gebruikt wanneer een snelle respons essentieel is. Interactieve gebruikers geven de voorkeur aan een snelle respons, maar real-time gebruikers eisen dit zelfs. Voorbeelden van real-time verwerking: controlesysteem voor het luchtverkeer op een vliegveld, robots, ...

Processen

Definitie proces:

Eén of meer reeksen opdrachten die door een besturingsprogramma worden beschouwd als een werkeenheid



Processen

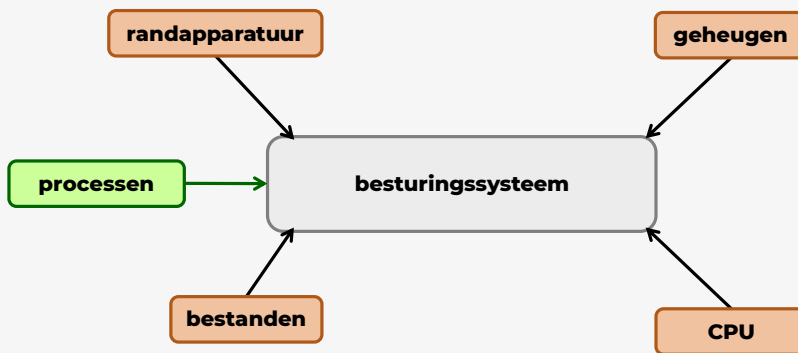
Een besturingssysteem moet reageren op alle onafhankelijke activiteiten die kunnen voorkomen. We noemen een dergelijke activiteit een proces. Een programma of taak bestaat uit één of meerdere processen. Een proces bestaat uit één of meer reeksen opdrachten die door een besturingssysteem worden beschouwd als een werkeenheid. Elk proces is dus een onafhankelijk uitgevoerde entiteit die meedingt naar het gebruik van bronnen (resources).

Het aantal programma's per gebruiker en het aantal processen per programma varieert. Eenvoudige programma's bestaan meestal maar uit één proces, terwijl complexere programma's vaak uit meerdere processen bestaan, en het aantal processen kan variëren in de tijd.

Een besturingssysteem bekommert zich typisch niet om de gebruiker en ook niet om het programma, maar focust op de uitvoering van de processen. Het besturingssysteem bepaalt dus welke processen momenteel uitgevoerd worden (runnen) en welke bronnen toegekend worden aan welke processen.

Resources

In eerste instantie spreken processen resources (bronnen) aan



**HO
GENT**

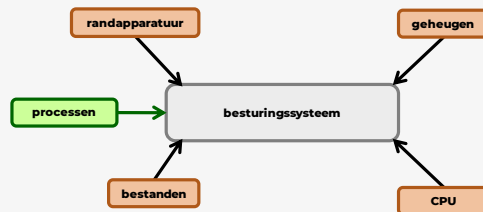
Resources

Het besturingssysteem moet open staan voor de behoeften van een proces. In de eerste plaats spreken processen resources (=bronnen) aan. Deze resources omvatten persistente opslag (bestanden), RAM geheugen, uitvoeringstijd op de CPU en eventuele communicatie met randapparatuur. Het besturingssysteem moet er dus voor zorgen dat elk proces toegang krijgt tot de nodige resources.

Resources

Een besturingssysteem moet:

- zorgen voor voldoende **geheugen** voor het proces
- het gebruik van de **CPU** regelen
- de gegevensstroom regelen van of naar **randapparatuur**
- **bestanden** en records kunnen lokaliseren



**HO
GENT**

Resources

- **Geheugen:** Een proces heeft geheugen nodig waarin het zijn instructies en gegevens kan opslaan. Een besturingssysteem moet er daarom voor zorgen dat het proces voldoende geheugen krijgt. Geheugen is echter een eindige resource. Het besturingssysteem mag niet toelaten dat een proces zoveel geheugen heeft dat de andere processen niet kunnen “runnen”. Bovendien stellen privacy en beveiliging de eis dat een proces niet in staat mag zijn zich eigenmachtig toegang tot het geheugen van een ander proces te verschaffen. Het besturingssysteem moet deze resource niet alleen toewijzen, maar ook de toegang regelen.
- **CPU:** De CPU is ook een resource die elk proces nodig heeft om zijn instructies te kunnen uitvoeren. Aangezien er gewoonlijk meer processen dan CPU's zijn, moet het besturingssysteem het gebruik van de CPU regelen; dat moet echter wel eerlijk gebeuren. Als het goed is, krijgen belangrijke processen de CPU snel ter beschikking en mogen minder belangrijke processen de CPU niet gebruiken ten koste van andere.
- **Randapparatuur:** Randapparatuur of devices zijn onder andere printers, tapedrives en diskdrives. Net als bij de CPU zijn er gewoonlijk meer gebruikers dan devices. Wat gebeurt er als verscheidene processen naar dezelfde printer of dezelfde drive

willen schrijven? Het besturingssysteem moet uitzoeken wie tot wat toegang heeft. Het moet ook de gegevensstroom regelen wanneer de processen van devices lezen of naar devices schrijven.

- **Bestanden:** Het besturingssysteem wordt verondersteld snel een bepaalde file te kunnen lokaliseren. Ook wordt verwacht dat het snel een bepaald record in die file kan lokaliseren. Omdat opslagmedia vaak vele duizenden files bevatten en een file vele duizenden records kan bevatten, is dit een ingewikkelde taak.

Scheduling

Bij multitasking, en in het bijzonder multi-user systemen is **scheduling** een heel belangrijk concept.

Scheduling verwijst naar de manier waarop aan processen **prioriteiten** worden gegeven, vaak in combinatie met een prioriteitenwachtrij.

Scheduling komt later uitgebreid aan bod.



**HO
GENT**

Scheduling

Bij multitasking besturingssystemen, en in het bijzonder multi-user besturingssystemen is scheduling een heel belangrijk concept. Bij een single-tasking systeem is er maar één proces tegelijk actief. Bij een single-user systeem hoeft het besturingssysteem slechts de behoeften van één gebruiker te vervullen. In een multi-user-computer gaat het om de behoeften van velen. Dit kan moeilijk of zelfs onmogelijk zijn. Aangezien veel programma's de resources van de computer gemeenschappelijk moeten gebruiken, moet het besturingssysteem beslissen wie wat krijgt en wanneer.

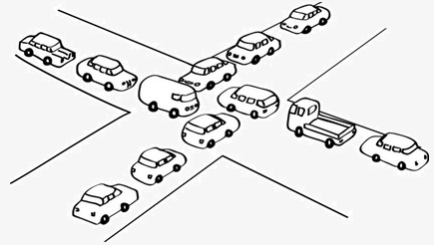
Vaak zal het besturingssysteem aan elk programma, of zelfs aan elk proces een bepaalde prioriteit toekennen. Maar hoe beslist het besturingssysteem welke prioriteit een bepaald programma krijgt? Elke gebruiker denkt dat zijn of haar programma de hoogste prioriteit zou moeten hebben en de programma's van alle anderen een lagere. Hoe lost het besturingssysteem dit probleem op? Hoe stelt het vast welke programma's werkelijk een hoge prioriteit verdienen?

Scheduling verwijst naar de manier waarop het besturingssysteem deze prioriteiten

toekent, en komt later uitgebreid aan bod in deze cursus.

Concurrency

processen zijn meestal niet onafhankelijk,
processen zijn concurrent



Bijvoorbeeld:

- 2 processen willen dezelfde printer gebruiken
 - 2 processen willen dezelfde file lezen of schrijven
 - 2 processen willen communiceren
- = **conflicten!**

OS regelt in welke volgorde processen afgehandeld worden
= **synchronisatie** (zie verder)

**HO
GENT**

Concurrency

In de meeste moderne systemen kunnen verschillende processen gelijktijdig uitgevoerd worden. Dit verhoogt de productiviteit, maar kan ook voor problemen zorgen. Wanneer 2 of meer processen gelijktijdig uitgevoerd worden en geen gemeenschappelijke bronnen gebruiken is er meestal geen probleem. De moeilijkheden ontstaan echter wanneer de processen gedeelde bronnen zoals gemeenschappelijke geheugen aanspreken. In dit geval kunnen er conflicten ontstaan. Het besturingssysteem zal deze dus moeten detecteren en indien mogelijk oplossen, bijvoorbeeld door een volgorde op te leggen waarin processen afgehandeld worden (synchronisatie). Concurrency en synchronisatie komen later nog uitgebreid aan bod.

Ontwerp-criteria

- Consistentie
- Flexibiliteit
- Overdraagbaarheid



Vaak is het onmogelijk om aan alle criteria te voldoen en worden sommige opgeofferd ten gunste van het andere.

**HO
GENT**

Ontwerp-criteria

Een modern besturingssysteem moet dus met heel veel zaken rekening houden. Het is duidelijk dat het beheer van resources en de uitvoer van de functies daarvan belangrijke criteria zijn bij het ontwerpen van besturingssystemen. Ook gebruiksgemak is een criterium.

Enkele belangrijke ontwerp-criteria voor een besturingssysteem zijn:

- **Consistentie:** Consistentie is een belangrijk ontwerp-criterium. Als het aantal processen, dat van de computer gebruik maakt, vrijwel constant blijft, hoort dat ook voor de respons te gelden.
- **Flexibiliteit:** Een besturingssysteem hoort zo te zijn geschreven dat een nieuwe versie het draaien van oude applicaties niet onmogelijk maakt. Bij een besturingssysteem moeten ook eenvoudig nieuwe randapparaten kunnen worden toegevoegd.
- **Overdraagbaarheid:** Dit houdt in dat het besturingssysteem op verscheidene soorten computers werkt. Overdraagbaarheid geeft de gebruiker meer flexibiliteit.

Al deze ontwerp-criteria zijn belangrijk; helaas is het meestal onmogelijk om aan alle

te voldoen. Vaak moet het ene criterium worden opgeofferd ten gunste van het andere. De ontwerpers moeten de omgeving waarin het besturingssysteem werkt, grondig kennen. Op die manier kunnen ze bepalen welke criteria voor de gebruiker het belangrijkste zijn.