

# Operating Systems : Hoofdstuk 1 - Besturingssystemen

## Warning

Een besturingssysteem (*Operating System* = OS) is een programma dat het mogelijk maakt de hardware van een computer te gebruiken.

Een gebruiker geeft geen instructies aan de hardware, maar aan het besturingssysteem. Een besturingssysteem geeft de hardware de opdracht de gewenste taken uit te voeren.

## Taken van een besturingssysteem

- Programma's de mogelijkheid geven informatie op te slaan en terug te herhalen
- 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

## Enkele voorbeelden

	Windows	Apple	Andere
x86 / x64 laptop/desktop	Windows 10 Windows 11 Windows Server 2022	MacOS 14 Sonoma	Fedora 39 Ubuntu 22/04 LTS AlmaLinux 9 <a href="#">[1]</a> ...
Mobile devices	Windows 10 Mobile (end of support)	IOS 17 iPadOS 17	Android 14 Plasma Mobile Ubuntu Touch ...

	Windows	Apple	Andere
ARM devices	Windows 11 on 1ARM	MacOS 14 Sonoma (for Apple silicon)	Rapsberry Pi OS Fedora 39 ARM Ubuntu 22 Server ARM AlmaLinux 9 ARM ...

## Soorten besturingssystemen

We kunnen een onderscheid maken tussen:

- *Single-tasking systemen*: 1 taak kan tegelijkertijd actief zijn
- *Multitasking, single-user systemen*: meerdere taken kunnen tegelijkertijd actief zijn
- *Mult-user systemen*: meerdere gebruikers kunnen gelijktijdig het systeem gebruiken

### Single-tasking besturingssystemen

#### Warning

Bij *Single-tasking* besturingssystemen is er maar één gebruiker die maar één applicatie tegelijk draait.

Een bekend voorbeeld hiervan is MS-DOS, de (niet-grafische) voorloper van Windows

### Multitasking single user besturingssystemen

#### Warning

De meeste besturingssystemen zijn multitasking, meerder programma's kunne tegelijkertijd geopend en uitgevoerd worden.

Deze systemen staan nog steeds maar één gebruiker tegelijkertijd toe. Maar deze kan wel verschillende taken simultaan uitvoeren.

Aangezien bij deze systemen een gebruiker verschillende taken tegelijkertijd uitvoerd, worden bepaalde functies van het besturingssysteem (zoals geheugenbeheer) ingewikkelder.

### Multi-User besturingssystemen

### Warning

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

## Concepten van besturingssystemen

---

### verschillende lagen

Veel besturingssystemen implementeren de interface tussen gebruiker en computer als een reeks stappen of lagen.

1. De bovenste laag of **shell** (of command interpreter) is de laag waarin de functies vastgelegd zijn. De gebruiker communiceert met deze laag. Deze laag bestaat uit routines van het besturingssysteem die ontworpen zijn om op opdrachten van de gebruiker te reageren.
2. Vaak vraagt een opdracht van een gebruiker veel van het besturingssysteem en zijn deze te ingewikkeld. Deze worden vaak niet door de shell uitgevoerd. De **utilities** laag bevat vele routines die deze zaken behandelt.
3. De onderste laag is de **kern of kernel**. Dit is het hart van het besturingssysteem. Deze laag bevat de routines die het vaakst gebruikt worden en waar het meest op aankomt.

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

### Programma's / taken

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

- **Interactieve programma's:** Dit is een programma dat een gebruiker vanaf de terminal activeert. Over het algemeen zijn die korte opdrachten die het besturingssysteem vertaalt en dan actie onderneemt. Het systeem zet vervolgens een prompt-teken en laat de gebruiker een volgende opdracht invoeren. Dit kan blijven herhaald worden. De gebruiker werkt met het besturingssysteem als het ware op een conversatie-achtige manier, *interactieve mode* genoemd. Interactieve gebruikers verwachten snelle respons daarom geeft het besturingssysteem voorrang aan interactieve gebruikers.
- **Batch-programma's:** Een gebruiker kan opdrachten in een file opslaan en deze aan de batch queue (wachtrij voor batch-programma's) toevoegen. Batch-gebruikers verschillen van interactieve gebruikers omdat zij geen directe respons verwachten. Bij schedulling 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 voorkeur aan snelle respons, maar real-time gebruikers eisen dit zelfs. (bv. controlesystemen voor luchtverkeer, robots,...)

## Processen

### Warning

Een **proces** is één of meerdere reeksen opdrachten die door een besturingsprogramma worden beschouwd als een werkeenheid

Elk proces is 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 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 processen. Het bepaald welke processen momenteel uitgevoerd worden en welke bronnen toegekend worden aan welke processen.

## Resources

Het besturingssysteem moet open staan voor de behoeften van een proces. In de eerste plaats speken 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.

Een besturingssysteem moet:


- *zorgen voor voldoende **geheugen** het proces:*  
Een proces heeft geheugen nodig waarin het zijn instructies en gegevens kan opslaan. Echter is geheugen geen *eindige resource* daarom moet het besturingssysteem zorgen dat het proces niet zoveel geheugen inneemt zodat de andere processen niet meer kunnen runnen. Ook mag een proces niet in staat zijn eigenmatig toegang tot het geheugen van een ander proces verschaffen. **Het besturingssysteem moet de resource niet enkel toewijzen, maar ook regelen.**
- *het gebruik van de CPU regelen:* De CPU is ook een resource die elk proces nodig heeft om zijn instructies te kunnen uitvoeren. Aangezien er meer processen als CPU's zijn moet het besturingssysteem ook het gebruik van de CPU regelen. Belangrijkere processen krijgen de CPU snel ter beschikking, minder belangrijke processen mogen de CPU niet gebruiken ten koste van anderen.
- *de gegevensstroom regelen van of naar randapparatuur:* randapparatuur of devices zijn onder andere printers, tapedrivers, en diskdrives. Er zijn gewoonlijk meer gebruikers dan devices. Wat gebeurt er wanneer er verschillende processen naar dezelfde printer of dezelfde drive willen schrijven? Het besturingssysteem moet dan uitzoeken wie toegang heeft tot wat. Het moet de gegevensstroom regelen wanneer de processen van devices lezen of naar devices schrijven.
- *bestanden en records kunnen lokaliseren:* Het besturingssysteem wordt verondersteld snel een bepaald bestand te kunnen lokaliseren. Ook wordt verwacht dat het snel een bepaalde record in een bepaald bestand kan lokaliseren.

## Schedulling

### Warning

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

Vaak zal een besturingssysteem aan elk programma, of zelfs aan elk proces een bepaalde prioriteit toekennen. Maar hoe weet het besturingssysteem welk programma welke prioriteit krijgt?

 Schedulling is het concept dat deze prioriteiten toekent.

## Concurrency

In de meeste moderne systemen kunnen verschillende processen gelijktijdig uitgevoerd worden. 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 gemeenschappelijk 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*)

## Ontwerp-criteria

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

Enkel belangrijke ontwerp-criteria voor een besturingssysteem zijn:

- **!** **Consistentie:** Als het aantal processen, dat van een 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 eenvoudige nieuwe randapparaten kunnen worden toegevoegd.
- **Overdraagbaarheid:** Dit houdt in dat het besturingssysteem op verscheidene soorten computers werkt. Overdraagbaarheid geeft de gebruiker meer flexibiliteit.

### **Caution**

Al deze ontwerp-criteria zijn belangrijk. Helaas is het meestal onmogelijk om aan alle te voldoen. Vaak moet het ene criterium worden opgeofferd worden voor het andere. Ontwerpers moet kunnen bepalen welke het belangrijkste is voor de gebruiker.