

NN31545.1451

Instituut voor Cultuurtechniek en Waterhuishouding  
Wageningen**ASPECTEN van INFORMATIEVERWERKING****44**

Bestandsorganisatie, een inleiding

J.R. Maassen

Nota's van het Instituut zijn in principe interne communicatiemiddelen, dus geen officiële publikaties.

Hun inhoud varieert sterk en kan zowel betrekking hebben op een eenvoudige weersave van cijferreeksen, als op een concluderende discussie van onderzoeksresultaten. In de meeste gevallen zullen de conclusies echter van voorlopige aard zijn omdat het onderzoek nog niet is afgesloten. Bepaalde nota's komen niet voor verspreiding buiten het Instituut in aanmerking.

## **ASPECTEN van INFORMATIEVERWERKING**

**44**

De nota's handelende over Aspecten van Informatieverwerkingen bevatten inlichtingen over de ontwikkeling van de informatieverwerkingen binnen het Instituut. Naast meer concluderende en toelichtende beschouwingen wordt aandacht besteed aan het gebruik van programma's, programmapakketten en apparatuur. Tevens worden inlichtingen gegeven over praktijkervaring met en toepassingen van de informatieverwerkingen.

## I N H O U D

	blz.
1. INLEIDING . . . . .	1
2. INFORMATIE EN GEGEVENS . . . . .	1
3. ORDENING VAN GEGEVENS . . . . .	2
4. KETTINGEN EN TABELLEN . . . . .	3
5. VORMEN VAN BESTANDSORGANISATIE . . . . .	5
6. EIGENSCHAPPEN VAN ELK VAN DE VORMEN VAN BESTANDSORGANISATIE . . . . .	6
6.1 DE SEQUENTIELE BESTANDSORGANISATIE . . . . .	6
6.2 DE DIRECTE (RELATIEVE) BESTANDSORGANISATIE . . . . .	8
6.3 DE GE-INDICEERD SEQUENTIELE BESTANDSORGANISATIE . . . . .	9
7. OPSLAGMEDIA . . . . .	11
8. TOEPASSINGSPROGRAMMATUUR . . . . .	12
LITERATUUR . . . . .	14

## 1. INLEIDING

Voor een optimaal gebruik van een computersysteem is het noodzakelijk dat de gebruiker weet van welke mogelijkheden het betreffende computersysteem is voorzien. Dit geldt niet slechts voor ter beschikking zijnde programmatuur, maar evenzeer voor de opslag- en verwerkingsmogelijkheden van gegevens. Immers, programmatuur en gegevens dienen volkomen op elkaar afgestemd te zijn wil het produkt, informatie, juist zijn. Nu komt het helaas nogal eens voor dat de wensen van de gebruiker niet geheel stroken met de mogelijkheden van het computersysteem, waarvan gebruik wordt gemaakt. Een compromis zal dan moeten worden gevonden.

In deze nota zal wat verder op deze problemen worden ingegaan, met als doel enig inzicht te verschaffen in het kiezen van de juiste vorm van gegevensopslag in een gegeven situatie. Het zal duidelijk zijn dat de structuur van de gegevens geanalyseerd moet zijn, voordat tot enige vorm van gegevensopslag kan worden besloten.

## 2. INFORMATIE EN GEGEVENS

Wanneer we informatie willen verkrijgen dienen we ons wel te realiseren waarover we geïnformeerd willen worden. Hierbij doet zich al een probleem voor, want wat is informatie? Voor het antwoord op deze vraag wordt in de informatica veelal de volgende definitie gehanteerd:

INFORMATIE is het PRODUKT dat verkregen wordt na VERWERKING van GEGEVENS.

Om informatie te verkrijgen moeten dus gegevens worden verzameld en verwerkt. Dit levert meteen het volgende probleem op, want nu dient men zich af te vragen welke gegevens moeten worden verzameld en op welke manier de verzamelde gegevens moeten worden verwerkt. Het antwoord op deze twee vragen is niet zo gemakkelijk te geven. Informatie is meestal niet hetzelfde als de som van een aantal gegevens, netzomin als een cake de som is van meel, water en suiker.

Bovendien kan de waarde van informatie door verschillende personen verschillend worden beoordeeld. Wat voor de een als waardevol geldt is voor de ander niet meer dan een nietszeggende reeks cijfers. Zo zal de directie van een bedrijf geïnteresseerd zijn in omzetresultaten, terwijl de productiemedewerker meer interesse zal hebben voor de cijfers op zijn loonstrook. Het is duidelijk dat hier in beide gevallen van verschillende gegevens gebruik moet worden gemaakt.

Ook de interpretatie van informatie, ontstaan uit dezelfde gegevens, kan van persoon tot persoon verschillend zijn. Wanneer bijv. een recreant op een mooie zomerdag een onweersbui ziet naderen, zal hij van zijn fiets afstappen en regenkleding aantrekken, omdat het weer verslechtert. De boer, die slechts 100 meter verder woont, zal daarentegen blij zijn met de naderende regen, omdat hij zijn akkers niet hoeft te beregenen.

Uit bovenstaande voorbeelden moese blijken, dat zeer voorzichtig moet worden omsgesproken met het besrip informatie. Dit zal nos sterker het seval zijn indien de informatie m.b.v. een computersysteem is verkresen. Het verzamelen van sesevens zal inderdaad kunnen worden seautomatiseerd, evenals het bewerken van de sesevens. Echter, het beoordelen van de ontstane informatie kan beter niet aan de computer worden overselsten. Hoe zou een computer bijv. moeten weten wat soed of slecht weer is indien de mensheid zelf hieromtrent van mening kan verschillen?

### 3. ORDENING VAN GFGEVENS

Zoals we reeds zagen is informatie het produkt, dat wordt verkresen nadat sesevens op de een of andere manier zijn verwerkt. Er zal dus eerst vastgesteld moeten worden welke sesevens nodis zijn om de sewenste informatie te kunnen produceren. Het behoeft seen betoos, dat dit zorgvuldig dient te worden nagesaan. Wanneer later zou blijken, dat een bepaald seseven ontbreekt, of dat niet ter zake doende sesevens zijn verzameld, wreekt zich dit in onnodis verspilte tijd en kosten.

Wanneer vast staat welke sesevens nodis zijn om de sewenste informatie te kunnen produceren, is het verstandig deze sesevens nader te analyseren. Voor alles dient vastgesteld te worden wat de relatie is tussen de sesevens onderlins. Voor de analyse van de sesevens bestaan verschillende methoden. Voor een beschrijvins van een aantal methoden wordt verwezen naar Eilers (1982). Aan de hand van de analyse van de sesevens kan de sesevensstructuur worden bepaald. Hierbij blijkt vaak dat sommige sesevens een belangrijker functie hebben dan andere. Het is bijv. niet ondenkbaar dat in een sesevensbestand bepaalde sesevens ras bereikbaar zijn, nadat een ander seseven is selesen. We hebben dan meestal te maken met een sleutel (Eng.: key), die de toegang vormt tot dat bepaalde seseven.

Nadat de sesevensstructuur is vastseled kan worden bepaald in welke volgorde de sesevens eventueel sesorteerd moeten worden. Ordenins heeft tot doel het opzoeken van een seseven in een bestand te versnellen. Voorbeelden van ordenins uit het daselijks leven zijn het telefoonboek, de encyclopedie, de gouden sids.

Ordenins seschiedt naar een of meer sezichtspunten. Het telefoonboek is seordend naar woonplaats. Per woonplaats is het telefoonboek seordend naar achternaam en per achternaam naar straatnaam. De sevolsen van het niet ordenen zijn in dit seval duidelijk: het niet seordend zijn van het telefoonboek zou betekenen dat van elke naam of instelling nagesaan moet worden of dit de sezochte persoon of instelling is, waarbij voorsaan in het telefoonboek besonnen moet worden. Overisens is het telefoonboek niet slechts seordend naar woonplaats en achternaam, maar ook naar groepen. Zo vinden we het telefoonnummer van de huisarts niet alleen in het alfabetisch naaresister, maar ook onder de groep "huisartsen".

Hiermee zijn we bij een ander besrip sansekomen, n.l. redundantie. Redundantie wil zessen dat een seseven op meer dan een plaats is vastseled. Het zal duidelijk zijn dat redundantie, indien mogelijk, dient te worden bererkt. Het op meer dan een plaats vastlessen van sesevens kost immers ruimte en dus ook geld. Bovendien heeft redundantie consequenties bij het aanbrensen van veranderingen in een bestand.

Echter, in bepaalde gevallen zal redundantie niet te vermijden zijn. Wanneer bij een bewerking sesevens uit meer dan een bestand moeten worden geselecteerd, is het vaak zinvol in elk van de bestanden een seseven op te slaan dat als sleutel fungeert, bijv. een uniek nummer, een datum o.i.d. Het opzoeken van een seseven kan in dergelijke gevallen aanzienlijk worden versneld, terwijl de controle op de juiste werking van de toepassingsprogrammatuur wordt vereenvoudigd.

#### 4. KETTINGEN EN TABELLEN

In de informatica worden sesevens veelal geordend met behulp van lijsten. In het dagelijks leven wordt met het begrip lijst echter meestal een tabel bedoeld, bijv. een telefoonlijst of de ledenlijst van een vereniging. Ook in de informatica heerst (nog) geen eenduidigheid in spraakgebruik t.a.v. beide begrippen. Deze begripsverwarring is te wijten aan een aantal overeenkomsten tussen beide begrippen. Als voorbeeld kunnen we de namen van het personeel van twee willekeurige afdelingen van een bedrijf beschouwen. In figuur 1 zijn de namen van de personeelsleden in alfabetische volgorde weergegeven in de vorm van een tabel. Deze tabel stelt dus de hele verzameling personeelsleden in de twee afdelingen voor. Binnen de tabel kunnen we verschillende deelverzamelingen onderscheiden:

- de deelverzameling public relations;
- de deelverzameling administratie.

Door middel van kaders om de afzonderlijke namen is weergegeven tot welke deelverzameling elke naam behoort.

Aalbers	
-----	
* Arends	*
-----	
* Edens	*
-----	
Feenstra	
-----	
* Gorter	*
-----	
Krol	
-----	
Leeshwater	
-----	
Dosterhout	
-----	
* Versteeg	*
-----	

Legenda:

		afdeling public relations
-----		
*	*	afdeling administratie
-----		

Fig 1. Tabel waarin schematisch twee deelverzamelingen zijn weergegeven.

In figuur 1 was sprake van twee deelverzamelings. In de informatica wordt een deelverzameling aangeduid als een lijst. Een lijst wordt als volgt gedefinieerd:

Een LIJST is een herkenbare reeks selijksortige elementen, die echter NIET NOODZAKELIJKERWIJS in opeenvolgende lokaties van een geheusenaedium behoeven te zijn vastgelegd.

Een manier om een lijst te onderkennen is door middel van kettinsadressering. Kettinsadressering (Eng.: chaining) is een methode om elementen uit een verzameling met elkaar te verbinden tot een lijst, waardoor een deelverzameling ontstaat. Hiertoe wordt elk element uitgebreid met een extra veld, het kettinsveld of schakelveld, waarin een verwijzing is opgenomen naar het volgende element dat tot de deelverzameling behoort. Hierbij geschiedt de verwijzing naar het eerste element van de kettins vanuit een speciaal veld, het z.s.n. ankerveld (Eng.: anchor). Het einde van de kettins wordt door middel van een niet mis te verstaan teken in het kettinsveld van het laatste element van de betreffende lijst aangegeven. In figuur 2 zijn de namen van de personeelsleden van de twee afdelingen door middel van kettinsen weergegeven.

1	P.R.   201	<-----
	-----	- ankers
2	Adm.   202	<-----
	-----	
201	Aalbers   204	
	-----	
202	Arends   203	
	-----	
203	Edens   205	
	-----	
204	Feenstra   206	
	-----	
205	Gorter   209	
	-----	
206	Krol   207	
	-----	
207	Leeshwater   208	
	-----	
208	Oosterhout   *	
	-----	
209	Verstees   *	
	-----	

Fig 2. Twee enkelvoudige kettinsen. De laatste elementen van elke kettins bevatten in het kettinsveld het ster-teken als aanduiding voor het einde van de kettins.

De hierboven geschetste kettinsen worden 'one way list' genoemd, omdat de kettins slechts in een richting kan worden doorlopen. Wanneer wijzigingen in de kettins worden aangebracht, moeten de wijzers in de elementen voorafgaand en volgend op het betreffende element worden aangepast.

Bij een 'two way list' kan de kettins in twee richtingen worden doorlopen. Per kettins zijn dan twee kettinsvelden nodig: een veld voor het voorwaarts doorlopen van de kettins en een veld voor het achterwaarts doorlopen van de kettins. Bij wijzigingen moeten derhalve de wijzers in beide richtingen worden aangepast.

Behalve eindige kettinsen kent men in de informatica ook de gesloten kettins, de 'circular list'. Kenmerkend voor een gesloten kettins is de terugverwijzing vanuit het laatste element naar het anker van de kettins. Bij herhaald doorlopen van de kettins biedt dit voordelen t.o.v. de eindige kettins.

Voor een toepassing van het gebruik van kettinsen wordt tenslotte verwezen naar v. Gils en Vink (1983).

In plaats van alle elementen van een lijst uit te breiden met kettinsvelden kunnen de wijzers ook worden opgenomen in een tabel. Een dergelijke tabel wordt verwijzingsstabel (Eng.: pointer array) genoemd. Toepassingen van verwijzingsstabellen vindt men vooral in de omgeving van databanken. Bij gebruik van verwijzingsstabellen kan bij wijzigingen vaak worden volstaan met het wijzigen van alleen de tabel. Immers, wanneer een element niet langer tot de lijst behoort, hoeft alleen de verwijzingsstabel te worden veranderd. Overigens wordt hier opgemerkt dat een verwijzingsstabel pas achteraf wordt opgesteld, dit in tegenstelling tot de wijzers in kettinsen. In figuur 3 is een verwijzingsstabel weergegeven.

P.R.	201	204	206	207	208
Adm.	202	203	205	209	

Fig 3. Twee deelverzamelings vastgelegd d.m.v. een verwijzingsstabel.

## 5. VORMEN VAN BESTANDSORGANISATIE

Voor de opslag van gegevens bestaat een aantal mogelijkheden, waarvan elk zijn specifieke voor- en nadelen heeft. Afhankelijk van de eisen, die worden gesteld aan de toegang tot de gegevens, de snelheid waarmee de gebruiker de gewenste informatie moet verkrijgen en tenslotte de opslagmogelijkheden van het computersysteem, kan de manier van gegevensopslag worden bepaald.

Op de Vax-11/750 computer van het Staringgebouw staan de volgende mogelijkheden voor de opslag van gegevens ter beschikking:



- sequentieel. Bij deze vorm van bestandsorganisatie kan een willekeurig record pas worden bereikt nadat alle voorgaande records zijn geraadpleegd. Deze vorm van opslag kan dan ook het best worden gebruikt wanneer het grootste deel van het bestand bij de toepassing is betrokken. De toepassingsprogrammatuur zal derhalve beter niet interactief van karakter kunnen zijn.
- direct (relatief). Bij deze vorm van bestandsorganisatie kan een willekeurig record op elk gewenst moment worden bereikt. De toepassingsprogrammatuur gebruikt in dit geval slechts een klein gedeelte van het bestand, zodat postwijze verwerking m.b.v. interactieve programmatuur mogelijk is.
- se-indiceerd sequentieel. Bij deze vorm van bestandsorganisatie kan een willekeurig record op beide bovengenoemde manieren worden bereikt. Afhankelijk van de toepassing zal een groot of klein aantal gegevens worden geraadpleegd. Deze vorm van opslag is dan ook te beschouwen als een compromis (met alle nadelen van dien) tussen de beide bovengenoemde vormen van opslag.

In het volgende hoofdstuk zullen de eigenschappen van elk van deze mogelijke vormen van bestandsorganisatie worden besproken.

## 6. EIGENSCHAPPEN VAN ELK DER VORMEN VAN BESTANDSORGANISATIE

Afhankelijk van een aantal eisen, welke door de gebruiker zijn gesteld, zal gekozen moeten worden voor een van de hiervoor genoemde vormen van bestandsorganisatie. In dit hoofdstuk zal worden uiteengezet hoe men tot een verantwoorde keuze kan komen. Aan de hand van voorbeelden zal worden getracht duidelijk te maken waarom in de betreffende gevallen voor een bepaalde vorm van bestandsorganisatie is gekozen.

### 6.1 DE SEQUENTIELE BESTANDSORGANISATIE

Kenmerkend voor de sequentiele bestandsorganisatie is het feit dat een willekeurig record slechts kan worden bereikt nadat alle aan dat record voorgaande records zijn geraadpleegd. In een file van 250 records kan het 80-ste record dus pas worden gelezen nadat de 79 voorgaande records zijn gelezen. Men zou deze wijze van bestandsorganisatie kunnen vergelijken met het kamer na kamer of de deur lezen van de naam van de bewoner wanneer men niet weet in welke kamer of de naam de gezochte persoon zijn kantoor heeft.

Het zal duidelijk zijn dat deze wijze van bestandsorganisatie alleen dan zin heeft indien bij de verwerking van de gegevens het overgrote deel van de gegevens is betrokken, zoals dat bijv. het geval is bij tijdreeksen.

Ook de wijze waarop de gegevens worden verwerkt, postsewjs (real time) of groepssewjs (batch), is bepalend voor de keuze van bestandsorganisatie. Bij postsewijze verwerking dient de antwoordtijd (response time) immers zo kort mogelijk te zijn. Wanneer we bedenken dat de verwerkingstijd van een programma o.a. wordt bepaald door het aantal input/output handelingen zal moeten worden geconcludeerd dat een interactief programma dat gebruik maakt van een groot sequentieel bestand een onaantvaardbaar lange antwoordtijd oplevert wanneer steeds willekeurige gegevens uit de reeks apart moeten worden verwerkt.

Overigens wordt hier nog opgemerkt dat het al dan niet gesorteerd zijn van het bestand de zoektijd aanzienlijk kan beïnvloeden. In een gesorteerd sequentieel bestand is het gemiddeld aantal records dat gelezen moet worden voordat het record wordt gevonden dat aan het zoekargument voldoet gelijk aan  $(n+1)/2$ , waarbij  $n$  het aantal records van het bestand is. In een ongesorteerd sequentieel bestand ligt dat gemiddelde tussen  $(n+1)/2$  en  $n$ , afhankelijk van de kans dat het record niet wordt gevonden. Immers, bij niet vinden van het gezochte record is het aantal records dat gelezen wordt altijd gelijk aan  $n$ .

#### Voorbeeld 1.

In het kader van de Cultuurtechnische Inventarisatie van een gebied wil een gebruiker weten wat de gemiddelde kaveloppervlakte in dat gebied is. Omdat de gemiddelde kaveloppervlakte afhankelijk is van de oppervlakte van alle kavels in het betreffende gebied zal de gebruiker de beschikking moeten hebben over een kavelbestand waarin alle in het gebied voorkomende kavels zijn opgenomen. Bij deze toepassing ligt een sequentieel georganiseerd bestand voor de hand. Het al dan niet gesorteerd zijn van het bestand is hier niet van belang, omdat bij de berekening van de gemiddelde kaveloppervlakte alle kavels zijn betrokken.

#### Voorbeeld 2.

Wanneer bovengenoemde gebruiker behalve de gemiddelde kaveloppervlakte in het totale gebied ook de gemiddelde kaveloppervlakte per dorpsbehoren wil weten is het nog steeds zinvol het kavelbestand sequentieel te organiseren. Echter, nu is het wel van belang het bestand te sorteren naar dorpsbehorens. Immers, nadat alle kavelrecords van een dorpsbehoren uit het bestand zijn ingelezen kan de gemiddelde kaveloppervlakte in dat dorpsbehoren onmiddellijk worden bepaald. Nadat alle dorpsbehorens zijn verwerkt kan tenslotte de gemiddelde kaveloppervlakte in het gehele gebied worden bepaald.

## 6.2 DE DIRECTE (RELATIEVE) BESTANDSORGANISATIE

Kenmerkend voor de directe (relatieve) bestandsorganisatie is dat elk record op elk gewenst moment kan worden bereikt. Vanuit een programma kan een gewenst record dus rechtstreeks worden opgeroepen. Bij deze vorm van directe bestandsorganisatie wordt aan elk record een relatief computeradres toegekend. Het gevolg hiervan is dat de computer zelf weet waar ieder record zich bevindt, zodat een willekeurig record direct kan worden opgeroepen. Het zal duidelijk zijn dat dit in bepaalde gevallen voordelen biedt. Immers, bij de sequentiele bestandsorganisatie moet een record worden opgezocht en zoeken kost nu eenmaal tijd. Wanneer bijv. een magazijnbediende niet weet in welke gang en in welke stelling hij een bepaald artikel kan vinden zal hij het hele magazijn af moeten zoeken totdat hij het betreffende artikel heeft gevonden. Bij de directe bestandsorganisatie echter weet de magazijnbediende uit het hoofd (zijn chef heeft deze kennis niet nodig) wat het nummer van de gang en de stelling is zodat hij direct, dus zonder zoeken, naar de soe-de plaats toeloopt.

Directe (relatieve) bestandsorganisatie wordt veelal toegepast bij post-gewijze (real time) verwerking. Omdat een gewenst record niet hoeft te worden opgezocht in het bestand, maar eenvoudig kan worden opgeroepen, komt dit de antwoordtijd van het toepassingsprogramma ten goede. Ook het al dan niet gesorteerd zijn van het bestand is voor de verwerking op zich niet van belang. Vaak zal met het oog op controlemogelijkheden evenwel toch een of andere sortering worden aangebracht.

### Voorbeeld 3.

Een reisorganisatie heeft een groot aantal reizen in haar programma opgenomen. Elke reis wordt gekenmerkt door een uniek nummer. De reisorganisatie heeft bij al haar vertegenwoordigers een terminal geplaatst en al haar reizen opgenomen in een reizenbestand. Van elke reis is een record aanwezig waarin de voor een boekingskantoor relevante gegevens van elke reis zijn opgenomen. Per record komen o.a. de vertrekdata en het aantal niet gereserveerde plaatsen voor. Op een willekeurig moment komt een client het kantoor van een der vertegenwoordigers van de reisorganisatie binnenstappen en geeft te kennen dat hij een reis wil boeken. Het is hier van belang dat de client op een snelle wijze op de hoogte gebracht kan worden van de plaatsinssmogelijkheden op de door hem verkozen vertrekdatum. Het reizenbestand moet daarom direct toegankelijk zijn. Immers, alle reizen waarvoor de client een belangstelling heeft doen hier niet ter zake.

#### Voorbeeld 4.

Wanneer blijkt dat de sewenste vertrekdatum van de reis waarvoor de cli-ent uit voorbeeld 3 belansstellins heeft is volseboekt, kan hij of een an-dere vertrekdatum of een andere reis kiezen. In het eerste geval is dit geen probleem; omdat de sesevens van de betreffende reis reeds voor handen zijn. Indien de client echter besluit een andere reis te kiezen, zou bij sequen-tiele bestandsorganisatie het hele reizenbestand opnieuw moeten worden se-lezen, totdat de sewenste reis is gevonden. Bij de direct bestandsorganisa-tie kan het record waarop de sesevens van de nieuw sekozen reis voorkomen, eenvoudig worden opseroepen.

Wanneer een client op een sekozen vertrekdatum van een bepaalde reis kan worden seplaatst moet het reizenbestand onmiddellijk worden seuteerd. Het op een later tijdstip uitvoeren van mutaties kan hier n.l. vervelende con-sequenties hebben. Het is bijv. niet ondenkbaar, dat op een dag meer dan een boekinsen voor dezelfde reis en vertrekdatum plaatsvinden. Dit zou tot sevolgs kunnen hebben dat er meer plaatsen worden sereserveerd, dan er be-schikbaar zijn. Hier wordt dit voorkomen d.m.v. de postsewijze werkwijze, waarbij telkens het direct toesankelijke bestand wordt seraadpleesd en se-uteerd.

### 6.3 DE GE-INDICEERD SEQUENTIELE BESTANDSORGANISATIE

Kenmerkend voor de se-indiceerd sequentiele bestandsorganisatie is het feit dat een willekeurige record zowel sequentieel, als direct hereikt kan worden. Op het eerste gezicht lijkt dit de ideale vorm van bestandsorganisa-tie. Echter, we hebben hier niet slechts te doen met de voordelen van de twee bovensenoemde vormen van bestandsorganisatie, maar ook met de nadelen van beide vormen van bestandsorganisatie. Het werken met een se-indiceerd sequentieel bestand is vergelijkbaar met het zoeken in een woordenboek. In-sters, we kunnen een record "aanwijzen" en vervolgens sequentieel verder zoe-ken.

Bij de se-indiceerd sequentiele bestandsorganisatie wordt elk record sekenmerkt door een of meer sleutels (Eng.: key). Een sleutel is een unieke waarde, bijv. een grondsebruikersnummer. De computer houdt zelf tabellen (indices) bij, waarin (meestal) per schijvenpakket een cylinderindex aan-wezis is. Hierin wordt per cylinder bijsehouden wat de hoosst voorkomende sleutelwaarde is. Per cylinder wordt dan een spoorindex bijsehouden, waarin van elk spoor de hoosst voorkomende sleutelwaarde voorkomt. Aan de hand van deze indextabellen rekest de computer uit op welk spoor van welke cylinder het gezochte record zich bevindt. Het betreffende spoor wordt vervolgens "direct" seselecteerd en vervolgens sequentieel afsezoekt.

Bij de initialisatie van een se-indiceerd sequentieel bestand worden al-le records sequentieel inselezen, waarbij teselijkertijd de verschillende indextabellen worden aansemaakt. Wanneer nu op een later tijdstip een re-cord zou moeten worden verwijderd, wordt (meestal) slechts de bijbehorende sleutel uit de relevante index-tabel(len) verwijderd. Dit heeft tot sevolgs dat het betreffende record fysiek wel aanwezig, maar logisch niet meer toe-sankelijk is. Wanneer veel records moeten worden verwijderd is opnieuw ini-tialiseren van het bestand dus raadzaam.

Ook het naderhand toevoegen van records kan problemen opleveren. Dit te meer wanneer het om het tussenvoegen van records gaat. Omdat bij de initialisatie van het bestand alle records sequentieel worden wessgeschreven is fysiek tussenvoegen onmogelijk geworden. Immers, aan elke recordsleutel is een fysiek adres gekoppeld. Het tussenvoegen van records zou tot gevols kunnen hebben dat de indextabellen niet meer consistent zijn. Meestal zal het tussenvoegen van records om deze reden plaatsvinden in speciale overloopebieden. Op de plaats waar het record eigenlijk had moeten staan, vinden we dan een verwijzing naar het betreffende overloopebied. Bij het op deze wijze "tussenvoegen" van records wordt de consistentie van de verschillende indextabellen dus niet aangetast. Overigens vindt het wesschrijven van records naar overloopebieden sequentieel plaats, zodat de computer bij het teruglezen van een tussengevoegd record nogal wat handelingen moet verrichten. De tijd, benodigd om een later toegevoegd record terug te vinden, zal hierbij toenemen naarmate de overloopebieden vollopen. Het voordeel van het "direct" toegankelijk zijn van het bestand wordt hierdoor sterk afgezwakt.

Indien binnen een reeds bestaand record een veld moet worden gemuteerd, hangt de toegangstijd af van de plaats waar het record zich bevindt. Wanneer het record zich niet in een overloopebied bevindt, is de toegangstijd korter dan wanneer het record zich achteraan in een overloopebied bevindt. Het gemuteerde record kan vrijwel altijd op dezelfde plaats worden teruggeschreven.

Samenvattend kan worden gesteld dat de ge-indiceerd sequentiele bestandsorganisatie voordelen biedt wanneer het betreffende bestand:

- zowel voor stapelsgewijze als voor postsgewijze toepassingen moet worden gebruikt;
- een lase vervangingsgraad (Eng.: file turnover) heeft.

Opgemerkt wordt nog dat de vervangingsgraad, d.i. het percentage records dat per tijdseenheid wordt vervangen, hierbij een goede indicatie geeft over het tijdstip waarop het bestand dient te worden gereorganiseerd.

#### Voorbeeld 5.

In het personeelsbestand van een groot bedrijf bevindt zich van elk personeelslid een aantal gegevens. Achtereenvolgens komen per record voor:

- Personeelsnummer
- Naam
- Adres
- Woonplaats
- Burgerlijke staat
- Aantal kinderen
- Beroep
- Loonschaal
- Datum in dienst tredens

Het personeelsbestand is gesorteerd naar personeelsnummer. Het personeelsnummer funseert hierbij als sleutel. Immers, in het bestand heeft het personeelsnummer voor elk personeelslid een unieke waarde. Van de overige velden kan dit bepaald niet gezegd worden. Zo is het zeer wel mogelijk dat er 8 mensen in dienst zijn met de naam Jansen. Ook is het niet uitsloten dat een aantal personeelsleden hetzelfde aantal kinderen heeft.

Het bestand moet maandelijks in z'n geheel worden gelezen t.b.v. de salarisberekening van elk personeelslid. Bij deze toepassing is dus een sequentiele bestandsorganisatie gewenst.

Wanneer echter een personeelslid verhuist of in het huwelijk treedt moet het record van het betreffende personeelslid onmiddellijk worden gewijzigd. In dit geval is het niet zinvol het hele bestand in te lezen. Bij deze toepassing is een directe bestandsorganisatie gewenst.

Op grond van deze twee noodzakelijke toepassingen, waarbij gebruik wordt gemaakt van dezelfde gegevens, is het personeelsbestand geïndiceerd sequentieel georganiseerd.

## 7. OPSLAGMEDIA

Elk van de boven genoemde vormen van bestandsorganisatie stelt bepaalde eisen aan de opslagmogelijkheden van het computersysteem. Het gevolg hiervan is dat niet altijd de meest gewenste vorm van bestandsorganisatie kan worden toegepast. Voor elk van de vormen van bestandsorganisatie zal in het onderstaande worden nagesaan welke opslagmedia in aanmerking komen.

### SEQUENTIELE OPSLAG

Voor de opslag van sequentiele bestanden kunnen in het algemeen de volgende opslagmedia worden gebruikt:

- magneetband
- magneetschijf
- floppy disk
- magneetbandcassette
- ponsband
- ponskaart
- microfilm

De media ponskaart en microfilm zijn op de Vax computer van het Staringsgebouw niet in gebruik. Overigens kunnen de media magneetschijf en floppy disk ook voor de beide andere vormen van bestandsopslag worden gebruikt.

## DIRECTE (RELATIEVE) OPSLAG

Voor de opslag van direct toegankelijke bestanden kunnen in het algemeen de volgende opslagmedia worden gebruikt:

- magneetschijf
- magneetkaart
- floppy disk (in beperkte mate)

Het medium magneetkaart voor het opslaan van grote hoeveelheden gegevens is sterk verouderd en wordt derhalve vrijwel nergens meer toegepast. Voor wat betreft het medium floppy disk wordt hier opgemerkt, dat het meestal mogelijk is een willekeurig adres te kiezen, waarna sequentieel verder gezocht moet worden. Voor de opslag van direct toegankelijke bestanden op de Starinsvax biedt de magneetschijf derhalve de beste mogelijkheden.

## GE-INDICEERD SEQUENTIELE OPSLAG

Voor de opslag van ge-indiceerd sequentiele bestanden kunnen in het algemeen de volgende opslagmedia worden gebruikt:

- magneetschijf
- floppy disk (in beperkte mate)
- microfiche
- beeldplaat (nog in ontwikkeling)

Voor het medium floppy disk geldt, dat het meestal mogelijk is een willekeurig adres te kiezen, waarna sequentieel verder gezocht moet worden. De media microfiche en beeldplaat zijn op de Vax computer van het Starinsgebouw (nog) niet in gebruik, zodat de magneetschijf de beste mogelijkheden biedt voor de opslag van ge-indiceerd sequentiele bestanden.

## 8. TOEPASSINGSPROGRAMMATUUR

Reeds in een ander hoofdstuk werd gezegd, dat de wijze van bestandsorganisatie afhankelijk zal moeten zijn van de toepassings en de mogelijkheden van het computersysteem. Wanneer eenmaal voor de een of andere vorm van bestandsorganisatie is gekozen kan de toepassingsprogrammatuur verder worden ontwikkeld. In onderstaand schema is weergegeven welke soorten bestanden bij welke soorten toepassingsprogrammatuur passen.

Glohaal kan worden gesteld dat bij stapelwijze verwerking sequentieel georganiseerde bestanden in aanmerking komen, terwijl direct toegankelijke bestanden het meest geschikt zijn voor postwijze verwerkingsprogrammatuur.

Voor wat betreft de regels voor het op de Vax-11/750 computer van het Starinsgebouw openen en verwerken van elk der drie soorten bestanden in toepassingsprogrammatuur, wordt verwezen naar v. Doorne (1983).

bestand \ Programmatuur	stapelsgewijs	postsgewijs	beide
sequentieel	+	-	-
direct	-	+	-
ge-ind. sequentieel	-	-	+

Fig 4. Combinatiemogelijkheden van soorten programmatuur en bestanden. Een plus-teken duidt op een goede combinatie, een min-teken duidt op een minder geslaagde combinatie.



Literatuur

- EILERS, H.B., 1982. Systeemontwikkeling volgens SDM. Academic Service,  
Postbus 96996, 2509 JJ 's-Gravenhage.
- GILS, J.B.H.M. van en VINK, L.W., 1983. Cultuurtechnische Inventarisatie  
Glastuinbouw. Opschonen en presenteren van de basisgegevens en  
maken van de verzamelstabellen. ICW-nota 1413: pp. 20
- DOORNE, W. van, 1983. VAX-11 Fortran deel 2 (input en output). ICW-nota,  
Aspecten van Informatieverwerking 41 (in voorbereiding).