

BESTANDSORGANISATIE, ENKELE TECHNISCHE ASPECTEN*)

door W. Visser

1 Doel van bestandsorganisatie

Het werken met bestanden brengt met zich een bepaalde organisatie van deze informatieverzamelingen. Twee aspecten zijn daarbij van belang: de toegankelijkheid van de gegevens en de kosten die gepaard gaan met het gebruik. Bestandsorganisatie heeft dan ook als doel het zodanig ordenen van de gegevens dat een gevraagde toegankelijkheid tegen verantwoorde kosten wordt verkregen.

Maatgevend voor de toegankelijkheid van de gegevens in de informatieverzameling is de zoektijd, d.i. de tijd die nodig is voor het vinden van gegevens.

De gegevens zijn, gegroepeerd in records, in een geheugen vastgelegd. De zoektijd kan daarom worden gezien als de tijd die nodig is voor het vinden van de plaats van een record in het geheugen.

De kosten worden bepaald door de investeringen in programmatuur en apparatuur en de verwerkingstijden.

Alvorens in te gaan op de technieken van bestandsorganisatie, vermelden we eerst nog enige algemene begrippen.

2 Records, de bouwstenen van een bestand

De informatie vastgelegd in een bestand wordt ingedeeld in records. De records vormen de denkbeeldige „verpakking” van de informatie, op zodanige wijze dat deze geschikt is om door een computer te worden verwerkt. Ieder record bevat een groep gegevens. Een bestand kunnen we dan ook omschrijven als een verzameling records.

Een record is een min of meer zelfstandige verzameling van gegevens, die echter wel alle betrekking hebben op een bepaald onderwerp. Het onderwerp waarop de gegevens slaan, wordt in een record aangegeven door een „sleutel”. Uiteraard houdt deze definitie niet in dat alle gegevens die bij een sleutel behoren in één record moeten staan. Als sleutel, ook wel identificatiesleutel genoemd, kan bijvoorbeeld dienst doen een personeelsnummer, een boekingsnummer, of een gironummer.

De hiervoor gebezigde uitdrukking „min of meer” behoeft enige toelichting. Een computerprogramma kan niet meer dan één record tegelijk lezen of schrijven, wanneer daartoe opdracht wordt gegeven. Na een leesopdracht zal iets met het record moeten gebeuren, hetzij dat er direct bewerkingen mee worden uitgevoerd voordat de volgende leesopdracht wordt gegeven, hetzij dat het tijdelijk buiten het inleesgebied wordt opgeslagen.

Met betrekking tot de informatie die het bevat is een record veelal geen zelfstandige eenheid. De informatie behorende bij één sleutel kan om redenen van doelmatigheid in verschillende records zijn ondergebracht.

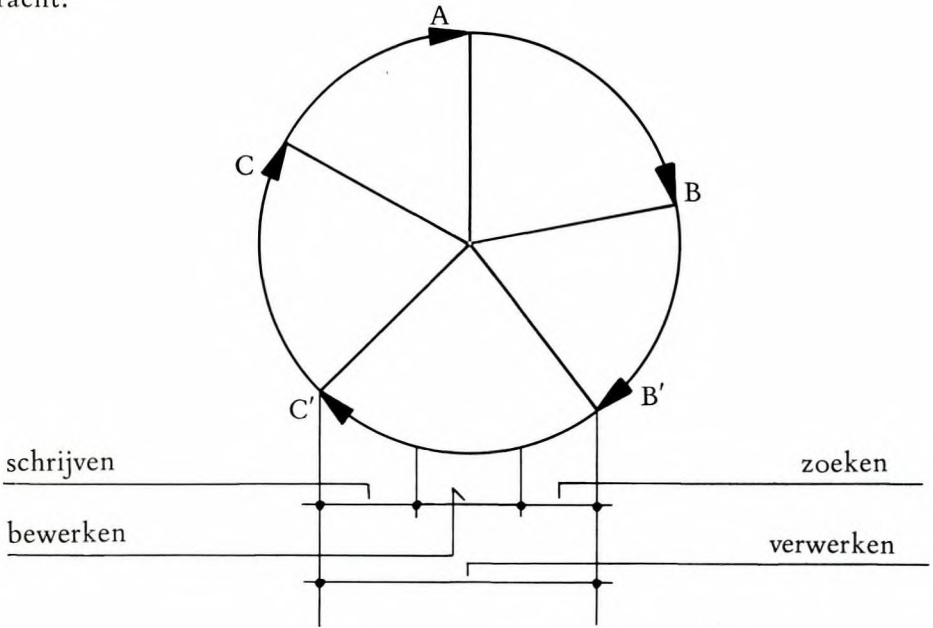
*) Het artikel is gebaseerd op ideeën, zoals neergelegd in het rapport BESTANDSORGANISATIE, deel 1 „De structurering van gegevensverzamelingen. Retrieval- en updateaspecten”, en de discussies die vooraf gingen aan dit rapport. Het rapport wordt uitgegeven door Uitgeverij Samsom, Alphen a/d Rijn, in samenwerking met het Nederlands Studiecentrum voor Informatica.

De gegevens zijn in de records opgeslagen in rubrieken, velden genoemd. De velden hebben een aan het programma bekende of herkenbare lengte en een bekende of herkenbare plaats. Door in het programma een veld bij de naam te noemen, kan men in het programma beschikken over de waarde, de gegevensinhoud, van het betrokken veld. Het programma kan daarna de vereiste bewerkingen uitvoeren.

Doordat de aard der gegevens, in verschillende records opgeslagen, verschillend is, verschillen ook lengte, plaats en namen van de rubrieken in verschillende records. Om onderscheid te maken tussen de verschillende records onderkent men recordsoorten.

3 Doorlooptijd en zoektijd

Onder de doorlooptijd wordt verstaan de tijd die verloopt tussen het geven van de opdracht aan het informatieverwerkend systeem en het gereed melden van de opdracht, eventueel onder aanbieding van de resultaten van de opdracht.



- A - de opdracht wordt verzonden
- B - de opdracht wordt aangeboden
- B' - het systeem start met de verwerking
- C' - het systeem beëindigt de verwerking
- C - de opdracht wordt gereedgemeld en de resultaten worden verzonden
- A - de resultaten staan ter beschikking

Afbeelding 1. Schematische voorstelling van het begrip doorlooptijd. In de afbeelding ontstaan de wachttijden B-B' en C'-C door stapelvorming. Sommige genoemde intervallen hebben in de praktijk in tijdseenheden gemeten de waarde nul.

Zoals uit afbeelding 1 blijkt, wordt de tijd die onder meer met behulp van technieken van bestandsorganisatie valt te bekorten, aangegeven door het interval B'-C'. Dit interval, de verwerkingstijd, bestaat voor een groot deel uit de tijd, nodig voor het zoeken naar de plaats van de benodigde gegevens, de zoektijd. De mate waarin deze zoektijd deel uitmaakt van de totale verwerkingstijd is afhankelijk van de bestandsorganisatie.

De analyse van het informatiesysteem waarvoor een geautomatiseerde uitvoering wordt gezocht, moet in ieder geval een inzicht geven in de wenselijke verwerkingstijden en de toelaatbare spreiding daarin, zodat de systeemontwerper een richtlijn heeft voor de eisen die aan de organisatie van de bestanden moet worden gesteld.

4 Technieken ter verkorting van de zoektijd

4.1 Splitsing van grondgegevens

De verkorting van de zoektijd kan worden bereikt door beperking van het aantal gegevens in een bestand. Voorbeelden hiervan zijn splitsing naar de mate van gebruik of splitsing naar de aard van de verschillende soorten grondgegevens.

Verschil in de mate van gebruik der gegevens, gekwantificeerd door de toegangsfrequentie der records, kan leiden tot splitsing van een informatieverzameling in verschillende bestanden, zonder dat de bestaande opbouw van de records in principe verandert. Te denken is hier bijvoorbeeld aan een productiebedrijf met een veelgebruikt bestand „orders in uitvoering”, en een weinig gebruikt bestand „uitgevoerde orders”. Nadat een order gereed is gemeld, worden de gegevens van de order van het eerstgenoemde bestand overgebracht naar het bestand met uitgevoerde orders.

Is de toegangsfrequentie op rubriekniveau uiteenlopend, dan kan zoiets leiden tot reorganisatie van de opbouw der records.

Het komt veel voor dat verschillende soorten informatie behorende bij één sleutel voor verschillende doeleinden worden gebruikt. Dit kan aanleiding zijn een informatieverzameling in verschillende bestanden onder te brengen, bijvoorbeeld de personeelsgegevens in:

- een bestand voor naam, adres en woonplaats;
- een bestand voor de salarisberekening;
- een bestand met gegevens omtrent opleiding en functie.

Naast de verkorting van de verwerkingstijd kan splitsing van bestanden eveneens worden ingegeven door andere overwegingen, zoals vertrouwelijkheid, geografisch spreiding, e.d.

4.2 Het structureren van bestanden

Door middel van het aanbrengen van een structuur in de gegevens, aangepast aan de eisen door het gebruik aan het bestand gesteld, kunnen interessante verkortingen van de zoektijd worden gerealiseerd.

Een bestand wordt gestructureerd door de records een vaste plaats, gekenmerkt door een adres, te geven, zodanig dat deze eenvoudig terug te vinden

zijn. De toewijzing van die vaste plaats in het bestand geschiedt volgens een bepaalde methode; voor het zoeken naar die plaats kan van diezelfde methode gebruik worden gemaakt. Dit zoeken kan veelal nog worden versneld door de toepassing van speciale technieken.

In de volgende paragrafen zullen zowel bestandsstructuren als zoek- en adresseermethoden nader worden belicht.

5 Bestandsstructuren

Naar structuur zijn de volgende bestanden te onderscheiden:

- sequentieel toegankelijk;
- willekeurig toegankelijk.

In een *sequentieel toegankelijke* structuur worden de records opgeborgen in klimmende of dalende volgorde van een bepaalde sleutel. Indien dit principe strikt wordt doorgevoerd, moeten bij het tussenvoegen van een nieuw record alle volgende records worden opgeschoven. Bij de sequentieel toegankelijke bestandsstructuur bestaat geen absolute relatie tussen de sleutel en het adres van de plaats op het geheugenmedium. We kennen alleen de plaats met betrekking tot de andere records. Deze gebrekkige kennis is van invloed op de methoden die we kunnen gebruiken om een bepaald record te vinden.

Het gebruik van de sequentieel toegankelijke bestandsstructuur gaat vaak samen met het groepsgewijs verwerken (batch-processing) van vragen en mutaties. Mutaties worden gedurende een bepaalde periode opgespaard (een uur, een dag, een week) teneinde op dezelfde wijze gesorteerd als het bestand, het oude bestand te muteren tot een nieuw bestand.

Anders dan bij een sequentieel toegankelijk bestand bestaat er bij een *willekeurig toegankelijk* bestand verband tussen de sleutel van het record en het adres. Hiermede wordt bereikt dat bij het zoeken naar een bepaald record de andere records niet onderzocht behoeven te worden. Aan de hand van de sleutel kan de plaats van het record direct worden bepaald. Geheugenmedia moeten echter adresseerbare gebieden kennen om geschikt te zijn voor deze bestandsstructuur.

De relatie tussen de sleutel en het adres kan op verschillende manieren worden gelegd. De eenvoudigste vorm bestaat daaruit, dat de waarde van de sleutel het adres zelf is. Een tweede manier om de relatie tot stand te brengen wordt gevormd door adrestabellen. Om het adres van een record te bepalen moet in een tabel de waarde van de sleutel van het betreffende record worden gezocht. In de tabel staat achter deze waarde het adres van het gebied waar het record staat. Zie ook paragraaf 6.5.

Een derde methode om de relatie tussen sleutel en het adres tot stand te brengen wordt gevormd door de „randomizing”-technieken. Hierbij wordt een bepaalde bewerking met de sleutel uitgevoerd leidende tot het adres van de fysieke plaats van dat record.

Vrijwel alle randomizing-technieken hebben als nadeel dat verschillende sleutels na omvorming hetzelfde adres kunnen opleveren, zogenaamde synoniemen, terwijl op dat adres slechts plaats is voor één of een beperkt aantal

records. Er zijn echter methoden ontwikkeld om te voorkomen dat synoniemen problemen veroorzaken.

Het vinden van de meest geschikte randomizing-procedure wil nog wel eens problemen geven. Veelal wordt het resultaat van enkele alternatieven tegen elkaar afgewogen en beoordeeld aan de hand van de verhouding tussen niet bezette adressen en het aantal records van het bestand, en aan de hand van het teveel aan synoniemen dat ontstaat.

6 Zoekmethoden

6.1 *Lineair zoeken*

Lineair zoeken houdt in dat beginnend bij het fysieke begin alle records van een bestand een voor een worden gelezen, totdat het gewenste is gevonden.

Lineair zoeken is de enige zoekmethode voor bestanden die op een geheugenmedium staan dat geen adresseerbare gebieden kent. Om te voorkomen dat het zoeken per vraag te veel tijd in beslag gaat nemen, worden de vragen verzameld en gesorteerd in dezelfde volgorde als het bestand, zodat voor het beantwoorden van alle vragen, het bestand slechts eenmaal wordt doorlopen. Veelal kunnen dan tegelijkertijd mutaties worden aangebracht.

6.2 *Zoeken volgens de halveringsmethode (binair zoeken)*

Om het gezochte record te vinden wordt halverwege het bestand een record gekozen. Door vergelijking op de gevonden kenmerken wordt bepaald of het gezochte record zich in de boven- of benedenhelft van het bestand bevindt. Door de aangewezen helft dan opnieuw als een bestand te beschouwen kan de cyclus worden herhaald totdat het gezochte record is gevonden. De records dienen op opeenvolgende adressen in volgorde van het zoekkenmerk te zijn opgeslagen.

6.3 *„Skip-sequential” zoeken (hinkelen)*

Dit is een zeer eenvoudige zoekmethode, te gebruiken bij een sequentieel toegankelijk gestructureerd bestand, waarbij met bepaalde passen door het bestand wordt gesprongen, totdat men constateert dat men te ver is. De records die de laatste sprong omvatten worden daarna lineair afgezocht.

6.4 *Kettingadressering*

Bij kettingadressering gebruikt men hulpgegevens in de vorm van verwijsadressen die in ieder record zijn opgenomen. Records die door middel van deze verwijsadressen volgens een bepaald gezichtspunt met elkaar zijn verbonden, vormen tezamen een sliert of ketting.

Zulke kettingen kunnen naar voren verwijzend, terugverwijzend, of heen-en-terugverwijzend worden opgebouwd. Bovendien kan één record deel uitmaken van verschillende kettingen.

Kettingen kunnen in reeds bestaande structuren worden gebruikt om additionele onderlinge relaties tussen bepaalde records vast te leggen.

6.5 Adrestabellen

Adrestabellen geven de relatie weer tussen sleutels en adressen, of de relatie tussen een groep sleutels en een adres. Wanneer zoektabellen niet in hun geheel in het interne geheugen kunnen worden geplaatst, kan een hiërarchische opbouw worden gerealiseerd, waardoor via een „getrapt” zoekproces uiteindelijk het adres van een record wordt gevonden.

De tabellen kunnen worden toegepast bij willekeurig en sequentieel toegankelijke bestanden. Bij de laatste uiteraard slechts wanneer het opslagmedium in adresseerbare gebieden is ingedeeld.

6.6 Andere zoekmethoden

De genoemde methoden zijn alle afzonderlijk toepasbaar. Maar door combinaties toe te laten en hierin bovendien kennis te verwerken met betrekking tot de *informatiestructuur*, kunnen zoekstructuren worden gecreëerd die nogal complex zijn.