

	Documentnummer	L
	Distribution	

EWD235 - 0

Contractie en expansie.

Samenvatting. Twee operaties, "contractie" resp. "expansie" gedoopt, worden onderkend. Zij worden bestudeerd in hun samenhang met assemblage, adresarithmetiek, geheugenindeling en protectie. Een en ander lijkt er op te wijzen, dat het begrip "adresruimte" en deszelfs zinvolheid voor een hernieuwd kritisch onderzoek in aanmerking komen. NB. Als associatieve geheugens op grote schaal ter beschikking komen, verliest dit onderzoek een aanzienlijk stuk van zijn relevantie.

1. Open en gesloten adressering.

Ik spreek van een gesloten adressering, wanneer de objecten van een verzameling geïdentificeerd zijn door consecutief oplopende nummers, zeg bij nul te beginnen. Een gesloten adressering laat zich implementeren door een op consecutieve geheugenplaatsen opgeborgen tabelletje: de adresarithmetiek -dwz. de overgang van identificerend nummer naar ~fysisch adres- impliceert optelling van identificerend nummer en ~fysisch beginadres van het tabelletje.

Ik spreek van een open adressering wanneer de objecten van een verzameling geïdentificeerd zijn door (nonnegatieve) nummers, waarbij het maximale nummer (veel) groter is dan het aantal objecten in de verzameling. In principe is een open adressering op dezelfde wijze te implementeren als een gesloten, als we nl. de gaten in de tabel voor lief nemen. Ik ga echter van een open adressering spreken als we ons dit niet meer kunnen veroorloven en we naar andere technieken moeten omzien (hashed adressering, associatieve geheugens, naamlijsten die gescand moeten worden etc.) Opm. De openheid van een adressering is dus een gradueel begrip: hij kan zo'n klein beetje open zijn, dat we hem nog als gesloten kunnen beschouwen.

Uitgangspunt van dit onderzoek is de observatie dat een beroep op open adressering vaak zo kostbaar is, dat men zoekt naar wegen om het aantal malen dat dit moet gebeuren, te reduceren. Het is hier, dat de operatie's contractie en expansie een rol spelen.

2. De operaties "contractie" en "expansie".

De operaties "contractie" en "expansie" zijn elkaars inverse.

Gegeven een grote verzameling objecten, waarvan ik eenvoudshalve aanneem, dat zij hierin door een gesloten adressering geïdentificeerd zijn. We noemen dit even "de globale adressering". Indien in een zekere context slechts naar een relatief kleine keuze van deze objecten gerefereerd wordt, dan is de globale adressering betrokken op deze keuze een open adressering. We spreken van contractie wanneer de gekozen objecten ten bate van de genoemde context aansluitend hernummerd worden; deze aldus ingevoerde

Supersedes doc. nr.	L	N.V. PHILIPS' COMPUTER INDUSTRIE APELDOORN	Date
			Page

	Documentnummer	L
	Distribution	

EWD235 - 1

"locale adressering" is betrokken op deze keuze dan gesloten. De locale adressering is een exploitatie van de wetenschap dat de context in kwestie zich afspeelt in een gecontraheerd universum.

Gegeven een context die via een gesloten adressering elementen van een verzameling bespeelt, dan spreken we van expansie als we genoemde verzameling gaan beschouwen als deel van een grotere verzameling en daarmee de elementen als elementen uit een grotere verzameling. De overgang van de locale adressering naar een globale is veronachtzaming van de wetenschap, dat de genoemde context zich in het gecontraheerde universum afspeelt.

(Ik zal in nog iets ruimer verband van expansie spreken, nl. als onze objecten informatiehoeveelheden zijn, voor elk waarvan een aansluitend stuk geheugen ter beschikking moet worden gesteld. De overgang van identificeren door nummer naar identificeren door beginadres zal ik ook expansie noemen.)

3. De organisatie van een subroutinebibliotheek.

In deze sectie beschrijf ik een subroutineorganisatie die de kenners zal treffen als een verder doorgevoerde automatisering van de subroutineorganisatie van de EDSAC 1 (1949). De keuze is hierop gevallen omdat dit het eenvoudigste voorbeeld lijkt waarmee we kunnen illustreren hoe expansie en contractie functioneren.

We beschouwen een subroutinebibliotheek met een zeer groot aantal routines. Ik neem aan, dat de routines in de bibliotheek geïdentificeerd zijn door catalogusnummer CN en dat catalogusnummers in de totale bibliotheek een gesloten adressering vormen.

Subroutineteksten verwijzen als regel naar zichzelf en naar enkele andere routines in de bibliotheek. Men zou dit in termen van catalogusnummers kunnen doen; in plaats daarvan wordt ten bate van de formulering van de routine contractie gepleegd en verwijst de subroutinetekst in een gesloten locale terminologie naar zichzelf en die paar anderen. (In de EDSAC-organisatie heetten de locale namen "code letters", bij de ARMAC en de X1 hebben ze als "sluitletters" nog jaren doorgeleefd.) Bij elke routine hoort een zg. "expansietabel", die de locale terminologie vertaalt in catalogusnummers; de expansietabel bevat dus een rijtje CN-waarden.

Zodoende kan een subroutinetekst zelf geformuleerd worden op een wijze die onafhankelijk is van de bibliotheek waarvan hij ~~XIX~~ deel uitmaakt (dwz. onafhankelijk van de omvang van de bibliotheek en de hierin gebruikte catalogusnummers). De locale terminologie heeft geen groter onderscheidend vermogen nodig dan vereist voor het onderscheid tussen de routines waarnaar hij werkelijk verwijst. Als om een of andere reden herindeling van de bibliotheek met verandering van de catalogusnummers gewenst is, dan beperkt de hieruit voortvloeiende bijwerkplicht zich tot de goed geïsoleerde expansietabellen. Tenslotte biedt de expansietabel

Supersedes doc. nr.	L	N.V. PHILIPS' COMPUTER INDUSTRIE APELDOORN	Date Page

EWD235 - 2

de uitwendige referenties op een presenteerblaadje.

Nu het probleem van assemblage; opdat dit een probleem zij neem ik aan dat de bibliotheek veel te groot is om permanent in het werkgeheugen te bivakkeren; voorts neem ik aan dat slechts een kleine fractie van de bibliotheek in een programma wordt opgenomen.

Als nu benodigde teksten en bijbehorende expansietabellen onveranderd in het geheugen zijn gekomen, welke mechanismen hebben we dan nog nodig?

Om steeds de heersende locale terminologie te kunnen interpreteren veronderstel ik aan wezig een toegewijd register, bevattende het beginadres van de vigerende expansietabel. Daar vinden we echter in eerste instantie het catalogusnummer terwijl men, om een subroutine aan te kunnen roepen, beginadressen van tekst en expansietabel moet weten. Sinds invoer zijn deze gegevens bekend; doordat echter maar een klein gedeelte van de bibliotheek in het werkgeheugen is opgenomen, vormt het catalogusnummer hiervoor een uiterst open adressering.

Het antwoord hierop is contractie. Voor de in het programma opgenomen routines voert men een gesloten adressering in; laten we deze de "selectie-nummers SN noemen. De taak van het assemblageprogramma omvat nu

- 1) het toekennen van selectienummers aan op te nemen routines
- 2) het bijwerken van op te nemen expansietabellen, zodat elke entry (ook) het toegekende selectienummer bevat
- 3) het invullen van de zg. "selectietabel"; dit is een expansietabel voor het hele programma voor de overgang van selectienummer naar de beginadressen.

Assemblage presenteert zich hier dus als contractie; omdat dat een vervelende operatie is loont het de moeite dit eenmalig per programma te doen. Subroutineteksten zijn ongewijzigd gebleven, de expansietabellen zijn per entry uitgebreid.

4. Vermindering van diepte van indirecte adressering.

De overgang van de locale terminologie van een subroutinetekst naar fysisch adres kost nu tweemaal het raadplegen van een expansietabel

- a) het raadplegen van de expansietabel van de heersende subroutine geeft een selectienummer SN
- b) met deze SN-waarde raadplegen van de selectietabel van het programma geeft een fysisch adres.

Als deze tweevoudige expansie te tijdrovend geacht wordt staan ons ter bespoediging twee wegen open, nl. het kortsluiten van de eerste, resp. de tweede expansie!

EWD235 - 3

Methode A. Men laat het assemblageprogramma de subroutineteksten herschrijven door de lokale terminologie te vervangen door SN's. De expansietabellen hoeven dan tijdens uitvoering niet meer geraadpleegd te worden: onder controle van de door de programmatekst geleverde SN-waarde kan in de selectietabel meteen het adres gevonden worden.

Methode B. De expansietabellen van de subroutines worden uitgebreid, zodat elke entry nu ook de beginadressen bevat, waardoor het frequent raadplegen van de selectietabel vervalt.

Methode A, die het meest de kant van klassieke assemblage uitgaat, heeft als grootste voordelen

- 1) het toegewijde register dat verwijst naar het begin van de heersende expansietabel kan vervallen (inclusief de hiermee geassocieerde red- en heratelprocedures).
- 2) van elke subroutine komt het beginadres slechts 1 keer in de administratie voor (nl. in de selectietabel) wat herziening van een dergelijk adres wel vergemakkelijkt.

Het nadeel van methode A is echter, dat de programmatekst niet meer onveranderd in het geheugen staat maar in een vorm, die afhangt van het grotere geheel, waar deze tekst nu deel van uitmaakt. En hoe langer ik erover nadenk, hoe zwaarder ik die prijs ga vinden.

Het punt is, dat expansie structuur verduistert en daarom zou men liefst het resultaat van de expansieoperatie zo volatiel mogelijk houden. Voelt men zich desondanks toch geroepen om een resultaat van expansie te laten ~~XXX~~ beklijven, dan is dit minder schadelijk wanneer dit beklijven tot de tabellen beperkt blijft dan wanneer het expansieresultaat tot je laagste niveau, de programmatekst, diffundeert.

Ik moge hiervan een paar voorbeelden noemen..

Methode A vraagt om iets als een "linkage editor", die bij methode B praktisch vervalt.

Methode A maakt het onmogelijk om verschillende programma's van de zelfde subroutinetekst gebruik te laten maken.

Als men de ene overblijvende expansie versnellen wil, dan vraagt dit bij methode A om een stukje heel snel geheugen zolang als de selectietabel (en als dat er niet is -de selectietabel groeit door combinatie!- om een klein associatief geheugen, dat de relevantste beginadressen bijhoudt); bij methode B kan volstaan worden met een snel geheugentje groot genoeg voor een expansietabel.

20 juni 1968

E.W.Dijkstra