

Over pagina-administratie

Het idee is, om in elk programma voor elke in dit programma voorkomende trommelpagina een zg. TPV (=trommelpagina variabele) in te voeren. De TPV's komen in de stapel van het bijbehorende programma. Ik zie minstens twee klassen:

- a) opdrachten-pagina's
- b) array-pagina's.

De TPV's voor de opdrachten-pagina's van een programma zijn lokale variabelen van het buitenste blok; zij worden geïnitieerd door de vertaler/inlezer.

De TPV's van grote array's worden, hoger in de stapel, geïnitieerd door de array-declaratie en afgevoerd door de blok-verlating.

De bedoeling van de TPV's is, om snel antwoord te kunnen krijgen op de vraag "Kaatje ben je boven?". Wij zouden dit kunnen bewerkstelligen door in de kernen een volledige -gestrekte- inhoudsopgave van de trommel(s) bij te houden, maar met duizend pagina's per trommel kost dit ons, als we niet in een onbarmhartig geschuif willen vervallen, per trommel minstens 3% van de kernen per trommel, ook al gebruik je deze trommelruimte niet.

Nu we de TPV's onderbrengen bij de lokale variabelen van de programma's hebben we bereikt, dat er in de kernen niet meer TPV's onthouden hoeven te worden, dan trommelpagina's, waarin mogelijk interesse kan bestaan.

Een gevolg hiervan is dat de interne formulering van het object-programma er vrij ongevoelig voor is, welke trommelpagina's nu eigenlijk door dit programma bezet worden: dit staat nl. niet meer door het hele programma verspreid, maar (éénmalig) in de TPV's. Voor de huishouding op de trommel neem ik aan, dat dit voordelen biedt; het betekent wel, dat de wijze, waarop bibliotheekprocedures naar elkaar moeten kunnen refereren nu met enige zorg vervult (met "later zorg", om precies te wezen).

Naast de, in de programma's ondergebrachte, TPV's hebben wij ook nog de "inhoudsopgave van de kernen". Deze bevat een element per kernpagina, de omvang van de kernpaginatable kunnen we meteen instellen op maximaal gebruik van de kernen. (Gebruiken we een stuk van die inhoudsopgave niet, dan betekent dit nl. dat we een stuk van het kerngeheugen niet nodig blijken te hebben, maar dan hadden we het kennelijk breed en konden we het ook breed laten hangen.) De relatieve prijs van deze tabel is de omvang van een element/omvang van een kernpagina. Dit is weer een argument voor niet te kleine pagina's. Elementen van deze inhoudsopgave noem ik KIE (Kernen Inhoudsopgave Elementen).

We gaan nu onderzoeken, wat alzo in een TPV en een KIE onthouden moet worden. De tendens zal zijn, om de omvang van een TPV klein te houden -hiervan kunnen we er immers zoveel hebben.

In een TPV moeten we in de eerste plaats onthouden, of zich een copie in de kernen bevindt. Ik stel voor:

TPV \geq +0 betekent: pagina in kernen aanwezig

TPV \leq -0 betekent: pagina niet in kernen aanwezig.

In het geval TPV \geq +0 moeten we dus aangeven, waar deze pagina zich dan wel bevindt. We kunnen opgeven: het beginadres in de kernen of het adres van de overeenkomstige KIE (deze zijn uit elkaar af te leiden). Ik denk, dat het KIE-adres het handigste is, de KIE hebben we nl. waarschijnlijk toch nodig. (Ik laat nog in het midden, of we in de KIE wellicht het kernenadres van de pagina opnemen; dit is dan abundant, maar kon ons wel eens op een welkome wijze geschuif besparen. Zodra we dit doen, zijn we bovendien var. tweemachten af als opgelegde lengte van een KIE.)

In het geval TPV \leq -0 moeten we nog onderscheid maken tussen "trommelpagina gereserveerd" en "nog geen trommelpagina gereserveerd". Dit laatste kunnen we nl. tegenkomen bij een TPV, die hoort bij een groot array. In het eerste geval kunnen we de TPV het beginadres op de trommel laten bevatten, het tweede geval kunnen we b.v. karakteriseren door TPV = -0.

Referentie naar een trommelpagina met TPV \leq -0 loopt nu verschillend. In beide gevallen moet er een kernpagina gekozen worden. Als TPV \neq -0, dan is er een -nu een wel-omschreven- transportplicht van trommel naar kernen; als TPV = -0, dan moeten we uit de "vrije trommelpaginalijst" een nieuwe trommelpagina aanvragen, van deze "bezetting" nota nemen en niet feitelijk transporteren. (Deze transport is overbodig en misschien zelfs schadelijk: waarom zou je b.v. nodeloos eisen, dat bij lang niet gebruikte trommelpagina's de pariteit zou deugen?)

Opm. De toekenning van trommelpagina kan ook uitgesteld worden, totdat werkelijk gedumpt moet worden.

Nu gaan wij over tot de KIE. Als wij voor een TPV slechts één woord ter beschikking stellen, dan betekent dit, dat we in de TPV geen ruimte meer hebben om het trommeladres op te bergen, zodra de pagina zich op de kernen bevindt. M.a.w.: dan hebben we de plicht om in de bijbehorende KIE het trommeladres bij te houden.

Wat we in een KIE allemaal gaan bijhouden, weet ik nog niet zo precies; de KIE's zullen nl. ook uitgebreid door de coordinator bespeeld worden. Ik zie zo met het blote oog drie toestanden voor een kernpagina, toegekend, wisselend en vogelvrij.

Een kernpagina wordt vogelvrij, zodra er in de inhoud geen interesse meer is. Het treedt op bij blokverlating, als dit blok grote arrays bevat. Bij deze blokverlating moeten de TPV's, die hierdoor beven de stapeltop komen, gescand worden. Vinden we een TPV = -0, dan hoeven we niets te doen, vinden we een TPV < 0, dan moeten we de in de TPV vermelde trommelpagina aan de lijst van vrijen toevoegen; vinden we een TPV > 0, dan is dat een KIE-adres. De in deze KIE vermelde trommelpagina worde aan de lijst van vrijen toegevoegd en in de bijbehorende KIE moeten we nu "vogelvrij" noteren. Voor de opdrachtenpagina's treedt dit op bij definitieve beëindiging van het programma.

Het proces "wisselend" moeten wij waarschijnlijk separaat in de KIE aangeven: zodra een kernpagina een nieuwe rol heeft toebedeeld gekregen en deze toekenning impliceert trommel-transporten, dan mag de coördinator niet aan deze kernpagina komen: de coördinator mag aan deze kernpagina pas een andere rol toekennen als het transport voltooid is. (Hetzelfde respect moet de coördinator hebben voor pagina's, die als bufferruimte voor autonome transporten van en naar de communicatie-apparatuur fungeren.)

Wat zich in de KIE van een toegekende kernpagina moet bevinden, is wat beter te overzien.

Ten eerste moet in de KIE onthouden worden het bijbehorende beginadres op de trommel- we hebben immers aangenomen, dat er in de TPV van een aanwezige pagina daarvoor geen ruimte meer zou zijn.

Ten tweede moet de KIE een verwijzing naar de TPV bevatten zoals de TPV een verwijzing naar de KIE bevat. (We zijn immers bezig met een soort dubbele boekhouding.)

Als de coördinator een vrije kernpagina zoekt (of moet maken), dan neem ik aan, dat hij het kerngeheugen in zijn totaliteit beschouwt d.w.z. inspectie pleegt in de KIE-tabel (zie ook onder, bij de "look back" informatie). Als de coördinator dan gekozen heeft, welke kernpagina nu van rol zal veranderen, dan moet bij de aanvankelijk bijbehorende TPV de verwijzing naar deze KIE weggehaald worden. Op grond van de afspraak in de vorige alinea doet de coördinator dit, door aldaar, bekalve TPV negatief te maken, ook het trommeladres weer in te vullen. (Dit suggereert, om als KIE-onderdeel, dat in de vorige alinea genoemd is, zonder meer te kiezen: de TPV-waarde bij afwezigheid.)

Het is even de vraag, hoe we deze verwijzing naar TPV denken te doen. Een mogelijkheid is "fysisch adres". Dit geeft ons dan wel de plicht om bij stapel-verschuiving de KIE-lijst af te lopen, om te kijken welke adressen daardoor bijgewerkt moeten worden. Een andere mogelijkheid is, om in de KIE de TPV te localiseren ten opzichte van de stapelbodem. Tegen dit laatste lijkt me geen bezwaar. (Wijziging van KIE's komt toch alleen maar voor bij pagina-wisseling.) Welbetekent dit, dat we in de KIE moeten vermelden, bij welk programma de KIE hoort.

(We stappen hier even buiten de gezichtskring van elk individueel programma. Ik neem aan, dat de coördinator alle onder behandeling zijnde programma's kent onder een nummer, dat aan dit programma blijft voorbehouden, totdat het de machine uitgaat. Ik stel tevens dat hij -ook om andere redenen- bij een nummer de stapelbodem zal kunnen vinden.)

Verder zullen we in een KIE, die beantwoordt aan een toegekende kernpagina, moeten noteren, dat deze kernpagina niet tot een stapel behoort. (Stapelpagina's zijn onze vierde groep: ze moeten ook met respect behandeld worden, omdat ze in principe op de kernen blijven staan en slechts met zorg verschoven kunnen worden.)

Verder zullen we in een KIE moeten noteren, of de inhoud van de kernpagina soms identiek is met die van de overeenkomstige trommelpagina. Dit geldt nl. voor alle opdrachten-pagina's; het kan tevens gelden voor getallen-pagina's. (Een betrekkelijk lang con-

stant array hoeft nooit teruggeschreven te worden.) Dit compliceert de assignment aan het element van een groot array. Dat wordt, vrees ik, toch niet zo'n triviale handeling en het zetten van een bit "beschreven" kan er, denk ik, nog wel bij.

Tenslotte is KIE de aangewezen plaats om "look back" informatie in op te slaan, d.w.z. notitie bij te houden van getoonde interesse. Mochten we ooit er toe overgaan een begrip als "starheid" te introduceren dan is ook hiervoor de KIE de aangewezen plaats. Bij gebrek aan beter stel ik me voorlopig maar voor dat we een NA Naur-achtig interesse-getal in de KIE zullen noteren. (Deze vorm van "look back" volgt de geschiedenis van kernpagina's; zodra een pagina teruggaat naar de trommel verliezen we hem geheel uit het oog. We zouden getempteerd kunnen zijn, om dan in de TPV nog een of ander interessegetal bij te houden. Experimenten hebben aangetoond - voorzover experimenten dit kunnen aantonen - dat dit niet profijtelijk is: de strategie wordt dan nl. beïnvloed door een gebeuren dat te lang geleden is om nog actueel te zijn.)