

Over de IBM360.

Een eerste serie bezwaren geldt de functionele specificatie van de kanaalorganisatie. Deze is ontworpen met de bedoeling dat de CPU een keten commando's ter sequentiele afwerking aan een kanaal kan aanbieden. De hardware is echter slecht bruikbaar doordat:

- 1) als een kanaal een interruptie signaal naar de CPU zendt voordat een vorig signaal van dit kanaal door de CPU gehonoreerd is, het vorige interruptie signaal onder tafel raakt;
- 2) het onmogelijk is op een betrouwbare manier een commando aan te haken aan een keten, die door het kanaal in bewerking is: probeert de CPU dit, dan is onder omstandigheden niet meer betrouwbaar vast te stellen, of het kanaal dit nieuwe commando nu nog wel of juist niet meer heeft uitgevoerd. (Deze onmogelijkheid is het gevolg van het feit dat de lengte van de commandoketen is aangegeven door markering van de laatste schakel.)
- 3) als de uitvoering van een commando mislukt, dit feit wel gemeld wordt, maar het kanaal na deze mislukking het volgende commando in de keten gaat uitvoeren alsof er niets mislukt was.

Ten gevolge van deze tekortkomingen kan de commandoketen niet gebruikt worden zoals bedoeld was en staat het operating system voor anders vermijdbare morele haastsituaties. (Zij, die niet in operating systems thuis zijn, voelen dit waarschijnlijk als detailcritiek. Ik vind het gesignaleerde defect echter een alarmerende indicatie ten aanzien van de competentie van het ontwerpersteam: hier is geen kwestie van smaak of stijl, er is met een zinvolle bedoeling het een en ander ontworpen terwijl een simpele redenering aantoont, dat de geboden faciliteiten ontoereikend zijn.)

Een tweede serie bezwaren bestaat uit voorbeelden, dat de ontwerpers de programma's teveel belast hebben met het beheer van de componenten van de machine, in plaats van het programma het proces op een abstracter niveau te laten vastleggen, het specifieke componentenbeheer aan systeem/machine delegerend. Voorbeelden hiervan zijn de volgende.

- 1) Peripherieapparaten kunnen slechts bediend worden door ze te koppelen aan een kanaal en vervolgens aan dit kanaal de commando's te geven. En dat terwijl het kanaal logisch geen betekenis heeft, het peripherieapparaat wel.
- 2) Het rekenorgaan bevat een groot aantal registers, waarvan in de programmatext expliciet staat aangegeven, welke er in elke opdracht gebruikt worden. Dit heeft onaangename gevolgen:

2a) dat compilers met het probleem van "register allocation" geconfronteerd worden, wat leidt tot een aan vertaaltijd kostbaar optimaliseringsproces

2b) dat het soort statuswisseling, dat optreedt bij subroutineoproep en multiprogrammering onontkoombaar tijdrovend wordt door red- en herstelplichten van registerinhouden (nodig of niet!).

Hier is in het ontwerp microscopisch geoptimaliseerd, terwijl je macroscopisch de prijs vele malen betalen moet:

ad 2a) de tijdrovendheid van het vertaalproces is verantwoordelijk voor de behoefte aan "onafhankelijk voorvertaalde programmaonderdelen", welke combinatie de behoefte aan een zg. linkage editor gecreeerd heeft;

ad 2b) als qua tijd de schoen begint te knellen, moet je de subroutineoproep vervangen door een aangepaste copie van de subroutinetext, wat aanleiding geeft tot heel lange programma's, zo lang, dat hun assemblage "a major processing task" wordt (Asher Oppler, IFIP 1965). IBM wil de dan benodigde geheugens graag leveren!

3) In plaats van dat het programma informatie adresseert, moet het geheugen adresseren, hetzij in kernen, hetzij disks, tengevolge waarvan elk programma individueel belast is met een prive-organisatie van "overlay's" en transporten tussen langzaam en snel geheugen. Dit heeft vrij rampzalige consequenties:

3a) Standaardprogramma's moeten bestaan in verschillende versies, al naar behoefte aan kerngeheugen. De wens programma's van anderen te gebruiken (APT voor de 360 bv.) kan je dwingen om je installatie met minstens zoveel kerngeheugen uit te rusten. Evenzo: vergroting van het kerngeheugen geeft niet geruisloos -dwz. zonder herprogrammering- efficiencywinst.

3b) De enige manier, waarop verschillende programma's enkelvoudig in het kerngeheugen aanwezige gemeenschappelijke routines samen kunnen gebruiken eist, dat deze routines permanent in het kerngeheugen staan. Omdat je hier dan weer zuinig mee moet zijn, is de behoefte aan "system generation" geschapen, systeemaanpassing aan "problem mix", met alle ellende van dien: het is een tijdrovend proces en bovendien wil je het niet, want je "problem mix" kan immers veranderen!

3c) Het feit, dat programma's gedurende hun executie een consecutief, onverschuijbaar stuk kerngeheugen bezetten verzwaart de scheduling task onnodig en buiten proportie, het resulterende systeem blijft stroef hanteerbaar.

Kortom: inbeddingsdetails, waarvan door iets geraffineerdere hardware geabstraheerd had kunnen worden, eisen nu hun expliciete representatie in de programma's. Enerzijds verzwaart deze onnodige expliciete representatie de programmaopbouw, anderzijds schaadt deze premature fixering de souplesse, waarmee de installatie nu nog gebruikt kan worden.

Het ongeloofelijke is, dat deze machine nu geadverteerd wordt met het "wonderful operating system", dat zoveel lasten van de schouders van de programmeur afneemt! Men verdoezelt hierbij:

- a) dat een groot gedeelte van deze lasten door de hardware zijn ontstaan en in een beter ontwerp lichter of non-existent waren geweest;
- b) dat dit operating system een schrikbarende overhead impliceert (maar IBM wil je graag een sneller model uit de serie leveren);
- c) dat de lasten de facto niet zijn opgevangen, maar overgeheveld zijn naar de directie van het rekencentrum, die de machine moet dimensioneren en beheren;
- d) dat het maken van dit operating system een opgave is gebleken, die het programmerend vermogen van de fabrikant te boven is gegaan;
- e) dat het systeem een zo barokke monstrositeit is geworden, dat geen gebruiker zich meer aan aanpassing kan wagen.

Ten leste, zoals de 709-serie me altijd getroffen heeft als een wanhopige poging om tapes als back up store te gebruiken, zo treft de 360-serie me als disk units met begeleidende electronica. Het zou me niet verbazen, als de wat teleurstellende performance van de snellere modellen in feite neerkomt op het feit, dat het snellere model, zeg, vier maal zo snel op de armbeweging staat te wachten. Hoe ze uit deze vicieuze cirkel moeten komen is me niet duidelijk: deze discrepantie opvangen door multiprogrammering is van de kant van de CPU erg onaantrekkelijk, overgang op "one-head-per-track-disks" zou wel eens veel van de motivering van het huidige operating system kunnen ontzenuwen.