Spelletjes onder Linux mini-HOWTO

Michiel Buddingh $Date: 2003/02/08 20:41:49 $

Nu Linux steeds populairder wordt, komen er ook steeds meer spellen uit voor Linux. In dit document zal ik proberen de vraag of Linux wel zo'n geschikt OS voor spellen is te beantwoorden. Ik ga dit doen aan de hand van uitleg over hoe Linux werkt ten opzichte van Win'95. Enige kennis over computers is wel vereist. Mensen zonder deze kennis kunnen ook maar beter geen Linux gebruiken. :-) De schema's die ik gebruik om verschillende dingen uit te leggen zijn natuurlijk sterk vereenvoudigd. Dat is nodig om ze duidelijker en begrijpelijker te maken. Mensen die het beter weten moeten zich maar niks van deze schema's aantrekken. Mensen die het beter weten hebben deze HOWTO ook helemaal niet nodig.

Inhoudsopgave

1 Wat besturingssystemen doen.

In tegenstelling tot wat je meestal hoort als simpele omschrijving van een besturingssyteem, is een besturingssysteem geen 'programma waarmee je andere programma's kunt draaien'. Het zou best mogelijk zijn bijvoorbeeld een tekstverwerker te programmeren die zonder besturingssysteem zou kunnen werken. Een nadeel hiervan zou zijn dat je de computer opnieuw zou moeten opstarten als je een spreadsheet zou willen gebruiken, en dat multi-tasking onmogelijk zou zijn (behalve dan als je twee computers hebt)
Ik zal de taken van een besturingssysteem eens op een rijtje zetten:

- Het beheren van het geheugen (RAM, Harde Schijven,Diskettes e.a.).
- De hardware besturen.
- Toegang bieden tot de twee bovenstaande items aan programma's.
- Dit weer doen in een manier dat geen één enkel programma complete aanspraak heeft op hardware, geheugen of processor.(multitasking)

Zoals bekend zijn sommige besturingssystemen hier beter in dan andere ;-)

1.1 De Hardware Abstraction Layer.

De Hardware Absraction Layer (voortaan HAL) is een gedeelte van het besturingssysteem dat functies aan een programma aanbiedt die hardware-onafhankelijk zijn. Laat ik dit uileggen aan de hand van een voorbeeld.

Een programma wil weten of de gebruiker een toets heeft ingedrukt. Om dit te doen zijn ongeveer 30 I/O-instructies nodig naar het keyboard. Nu is toetsenbord-input iets wat door heel veel programma's wordt gebruikt, en dus biedt het besturigssysteem daar een functie voor aan die door alle programma's gebruikt kan worden.
Afhankelijk van het besturingssysteem en programeertaal typt de programmeur gewoon `getch()` of >`INT 16h`. Dit scheelt de programmeur in tijd en frustratie, het programma wordt er kleiner door, en de kans op fouten vermindert; iedereen gebruikt dezelfde functie, die als het goed is foutloos werkt. Het heeft nog een voordeel: als ik een toetsenbord op een USB of PS/2 - poort zou hebben aangesloten, zou de bovengenoemde tekstverwerker een probleem hebben als hij niet weet hoe hij die dingen moet aansturen. In Schema:

```
+---------+ +-----+  +--------+
+         + +     + +          +      programma's
+---------+ +-----+  +--------+
     |         |         |
+-----------------------------+
+                             +      Besturingssysteem (HAL)
+-----------------------------+
   /     \
++++      \
++++       ||||                      Hardware.
++++       ||||
           ||||
```

Er is nog een voordeel: Programma's die directe toegang tot de hardware gebruiken zijn in een ideale gelegenheid het besturingssysteem te laten crachen.

1.2 De HAL onder Windows '95.

Wat mij natuurlјk brengt tot Windows '95. Win '95 is een populair OS voor spellen. Om uit te leggen hoe het dit geworden is, moeten we terug in de tijd naar de tijd van MS-DOS.
Vroeger, toen computers nog op steenkool draaiden, werd op IBM compatible systemen het besturingssysteem MS-DOS gebruikt. Het had geen multitasking, of behoorlijk geheugenbeheer, maar het had een redelijke HAL die het merendeel van de PC-hardware ondersteunde. En als MS-DOS het niet ondersteunde, kon je direct naar de hardware schrijven. Toen de eerste spellen voor MS-DOS geschreven werden, bleek dat de enige 'HAL-correcte' manier voor het zetten van puntjes op het scherm, gruwelijk langzaam was. Zeker, het werkte op alle hardware, maar deze methode was **zo** langzaam dat het niet snel genoeg was om een taartpuntgrafiek binnen redelijke tijd op het scherm te krijgen, laat staan pacman.
Dus werd er direct naar de Hardware geschreven, en veel. In Schema:

```
+-------------------------------------+
+               Spel                  +
+-------------------------------------+
     |                   |    |
     |                   |    +--------+
     |                   |             |
     |                   |             |
+-----------------+      |             |
|     DOS         |      |             |
```

```
+-----------------+          |            |
     |      |              |            |
     |    +-----------------+----+        |
     |    |        BIOS         |        |
     |    +----------------------+        |
     |      |              |            |
+-----------------------------------------+
|                Hardware                |
+-----------------------------------------+
```

Toen Windows populair werd (zo rond versie 3), werden het merendeel van de programma's ineens voor Windows geschreven, niet omdat het zo'n geweldig OS was, of omdat het grafisch was, maar omdat het de enige manier was om al het geheugen van de computer rechtstreeks te gebruiken. Onder MS-DOS was dit niet mogelijk. Spelletjes voor Windows kwamen er alleen niet veel. Windows **stond** er namelijk op dat programma's de Windows-HAL gebruikten voor grafische functies. Deze HAL was niet geschreven voor spelletjes, maar voor tekstverwerkers en spreadsheets; Het was makkelijk om er een knop, of titelbalk mee te tekenen, maar voor spelletjes was het té langzaam.

Microsoft wilde natuurlijk dat er ook spelletjes voor Windows geschreven zouden worden, Dus begon het met het ontwikkelen van verschillende spelletjes-HALs, zoals WinG, en uiteindelijk, DirectX.

1.3 De HAL onder Linux.

De HAL onder Linux is een heel ander verhaal. Direct schrijven naar de Hardware is iets wat in Linux alleen bij wijze van uitzondering mag worden gedaan door programma's die het **echt** nodig hebben. de HAL in Linux ziet er dan ook zo uit.

```
+-------------------+ +------------+
+     Spelletje     + +  Programma +
+-------------------+ +------------+
          |                   |
+----------------------------------+
+             Kernel               +
+----------------------------------+
                   |
+----------------------------------+
+             Hardware             +
+----------------------------------+
```

En je raadt het al: De Linux-kernel is óók niet ontworpen voor spelletjes. Er bestaan al andere spelletjes-HALs voor Linux, maar er is nog geen 'standaard' voor Linux die alle gebieden dekt, dus niet alleen graphics, maar ook geluid, en 3d, zoals DirectX voor Win95 is.

2 Conclusie

Het zal inmiddels duidelijk zijn dat er geen **echte** reden is waarom spelletjes onder Linux langzamer zouden moeten lopen. Als er eenmaal een goede standaard-HAL beschikbaar komt (er zijn nu verschillende troon-pretendenten), zal Linux nét zo goed spelletjes kunnen draaien als Windows '95.
Tot die tijd kan ik mensen die hun computer alléén voor spelletjes gebruiken alleen maar aanraden om het bij Win95 of 98 te houden.

3 Nawoord

Deze mini-HOWTO werd oorspronkelijk geschreven als gewone html file door *Michiel Buddingh* (a.k.a. Ajuin). Later heeft *Gerrit Holl* het document vertaald naar sgml en een paar typefoutjes verbeterd.