VU-onderzoekers herontwerpen Posix besturingssysteem

Minix 3: veilig en betrouwbaar

De eerste versie van Minix is in 1987 aan de Vrije Universiteit in Amsterdam begonnen als open-source besturingssysteem voor onderwijs en onderzoek. Deze traditie is in Minix 3 voortgezet met een nieuw modulair en multi-server ontwerp dat specifiek gericht is op betrouwbaarheid. Met een microkernel, device drivers in user space, en automatische herstart van drivers na foutdetectie, profileert het OS zich ook voor embedded systems. "Ons doel is om een veilig en betrouwbaar besturingssysteem te ontwerpen; om aan tonen dat onze ideeën werken, brengen we dit ook in praktijk," zegt VU onderzoeker en medeontwerper Jorrit Herder.

HANS VAN THIEL

Softwarebugs zijn onvermijdelijk. Daarom moet een besturingssysteem zo worden ontworpen dat het effect van fouten beperkt blijft. Dat is de gedachte achter het vernieuwde ontwerp van Minix 3.

Minix is een besturingssysteem dat in 1987 door professor Andrew Tanenbaum en zijn medewerkers aan de Vrije Universiteit in Amsterdam is ontworpen voor onderwijs en onderzoek. Het is een Unix-systeem dat is gebaseerd op de Posix-standaard, maar het is klein genoeg om de basisprincipes van operating systems te demonstreren. Met het bijbehorende leerboek van Tanenbaum wordt Minix dan ook vooral in het wetenschappelijk onderwijs gebruikt. Om dezelfde redenen is Minix echter ook geschikt om nieuwe inzichten te implementeren en te testen.

De derde versie is dan ook heel anders opgebouwd, terwijl het naar buiten toe een Unix OS blijft. Dat betekent dat er in principe veel applicaties voor beschikbaar zijn. Minix 3 is ook weer open-source met een vrije BSD-licentie en wordt, behalve met een website, uitgebreid gedocumenteerd door een leerboek van Tanenbaum en enkele recente publicaties.

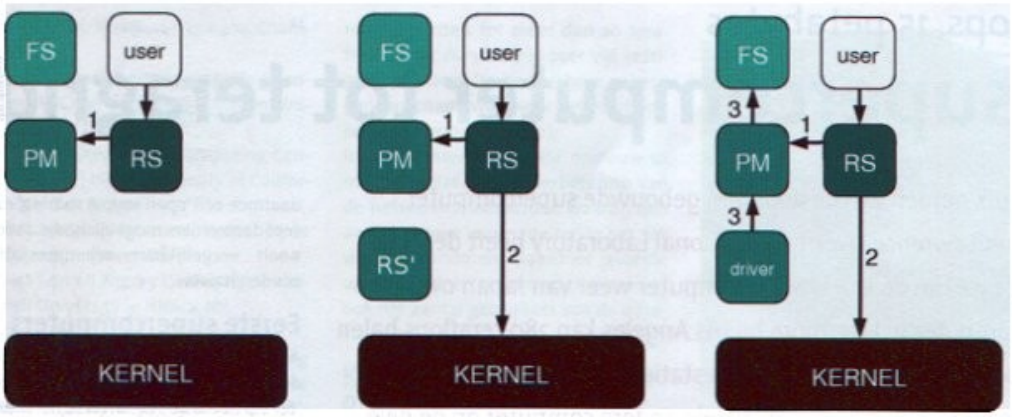
Compartimenten

"Het is te vergelijken met het plaatsen van waterdichte schotten in een schip," vertelt VU onderzoeker Jorrit Herder. Hij maakt als promovendus deel uit van de groep die de betrouwbaarheid van besturingssystemen tracht te verbeteren en in het kader daarvan Minix 3 ontwikkelt. Herder is net teruggekeerd van de 11th Asia-Pacific Computer System Architecture Conference in Shanghai, waar hij een presentatie, 'Reorganizing Unix for Reliability, heeft gehouden.'

"Als er dan toch ergens een lek ontstaat, dan kun je in ieder geval de gevolgen beperken." Uit onderzoek naar de foutdichtheid van software blijkt dat elke 1000 regels code 6 ... 16 bugs bevatten en dat zou voor de 2,5 miljoen regels Linux-kernel neerkomen op zo'n 25 000. Windows zou zelfs twee keer zoveel bugs met zich meedragen. Beide systemen vormen inwendig een eenheid en elke onvoorziene fout kan in principe tot een systeemcrash leiden.

Het is een illusie te denken dat software ooit foutvrij kan zijn. Systematische fouten kunnen vaak gevonden worden tijdens het testen, maar bugs die te maken hebben met timing, geheugengebruik, buffers, pointers, enzovoorts zijn lastiger op te sporen. Bovendien bestaat zo'n 70% van een besturingssysteem uit device-drivers en die zijn vaak afkomstig van derden. Het aantal fouten in device-drivers is drie tot zeven maal groter dan in de 'vaste' kern en drivers zijn dan ook verantwoordelijk voor de meerderheid van de crashes.

Daarom is er bij Minix voor gekozen de



Figuur 1. Stappen bij het starten van een nieuwe device driver: (a) De reincarnation server (RS) vraagt de process manager (PM) een nieuw proces voor de driver te maken; (b) de toegangsrechten van het nieuwe procss worden ingesteld bij de kernel; (c) de driver binary kan via PM en de file server (FS) worden uitgevoerd.

drivers te implementeren als user-mode processen die ingekapseld zijn in hun eigen adresruimten, beschermd door de MMU (memory management unit) van de hardware. Hierdoor wordt ook de eigenlijke kern kleiner (4000 regels in Minix 3) en dus overzichtelijker en eenvoudiger. Een dergelijke streng modulaire benadering is op zich niet nieuw, maar werd niet eerder in de praktijk gebracht vanwege de veronderstelde vermindering in prestatie. Metingen aan Minix 3 wijzen echter uit dat er weliswaar een verschil is, maar dat het snelheidsverlies onder de 10% blijft. Ook al omdat hardware nog steeds veel sneller wordt, wegen de toegenomen veiligheid en betrouwbaarheid daar ruim tegenop, aldus Jorrit Herder.

Multiserver Architectuur

De eigenlijke kern verzorgt interruptafhandeling, programmering van de CPU en MMU, proces-scheduling, en de feitelijke device-I/O. De IPC (Inter Process Communication) past een request-reply mechanisme toe zonder buffering en een IPC_NOTIFY voor asynchrone communicatie.

De kern gebruikt een aantal lijsten en bitmaps om privileges te handhaven. Dat zijn, bijvoorbeeld, IPC-bestemmingen, beschikbare kernel-calls, I/O-ports, IRQ-lijnen en geheugengebieden. Die 'policies' worden voor elk proces bepaald door een RS (Reincarnation Server) buiten de kern, maar worden gehandhaafd door de kern.

Die bevat slechts twee processen, die in Minix 3 'tasks' worden genoemd, SYS en CLOCK.

SYS is de interface voor alle user mode servers en drivers en alle kernel-calls in de system library worden vertaald naar request messages naar SYS. CLOCK is verantwoordelijk voor CPU-gebruik, proces-scheduling in de tijd, beheer van watchdog-timers en interactie met de hardwareklok. De diensten van deze taak verlopen via kernel calls naar SYS.

Al het overige wordt in Minix 3 geïmplementeerd door servers in user space. De afmetingen daarvan zijn ook weer beperkt, zo'n 1000 tot 3000 regels per module, zodat ze overzichtelijk blijven en gemakkelijk te onderhouden.

De process manager (PM) levert samen met de file server (FS) de eigenlijke Posix-interface voor applicaties. PM beheert processen en verzorgt ook de Posix signaalafhandeling. Een memory manager (MM) beheert voor alle processen geheugensegmenten en handhaaft de beveiliging. Op dit moment is er nog geen ondersteuning voor virtueel geheugen, maar daaraan wordt gewerkt. De file server (FS) werkt op dit moment alleen met het eigen Minix file system maar ook dat zal veranderen door ondersteuning van een virtueel bestandssysteem.

De data store (DS) is een kleine data base server met 'publish-subscribe' toegankelijkheid en DS kan niet alleen de toestand van een proces bewaren maar wordt ook gebruikt voor de toekenning en het beheer van globale identifiers.

Management

Het beheer van alle gepriviligeerde systeemservers en -drivers wordt uitgevoerd door de reincarnation server (RS). Al deze processen zijn forks van de RS (zie figuur 1) en als een proces termineert stuurt de PM een message naar RS. Bovendien kan, al naar gelang de policy, een statuscheck worden uitgevoerd. Als er een fatale fout optreedt in een component kan RS een nieuwe kopie opstarten en/of andere maatregelen nemen. De menselijke systeembeheerder kan policies specificeren in een shell script. Als een automatische herstart voldoet om de storing te verhelpen, dan kan die storing zelfs onzichtbaar blijven voor het aanroepende proces.

De toegenomen betrouwbaarheid van Minix 3 geeft niet alleen meer bescherming tegen fouten, maar ook tegen computervirussen en dergelijke. Hoewel daar tot nu toe niet de nadruk op heeft gelegen, is beveiliging dan ook een van de punten van verder onderzoek binnen de vakgroep, vertelt Jorrit Herder.

Omdat Minix 3 open source is kunnen ook geïnteresseerden van buiten de Vrije Universiteit er aan bijdragen. Het is gepoort naar de PowerPC en aan overdracht naar de XScale architectuur (met ARM core) wordt gewerkt. Met name vanwege de betrouwbaarheid, de grootte en de modulariteit, is Minix 3 in principe geschikt voor embedded toepassingen, aldus Herder.

"Onderzoek heeft niet zoveel zin als de kennis niet naar buiten toe wordt overgedragen. Het zou natuurlijk ook leuk zijn als dit werk een praktische toepassing zou krijgen. Maar onze belangrijkste doelstelling met Minix 3 is te demonstreren dat onze ideeën over besturingssystemen echt werken." ■

Bronnen:
"Reorganizing UNIX for Reliability," Jorrit Herder e.a., sep. 2006
"Construction of a Highly Dependable Operating System," Jorrit Herder e.a., okt. 2006
www.cs.vu.nl/~jnherder/publications.php
www.minix3.org