

Interfața driver-kernel la Linux

Introducere

Deși pentru unii dintre noi acest lucru poate fi o surpriză, cei mai specializați dintre noi în domeniul calculatoarelor știu că procesorul definit ca unitatea centrală de prelucrare nu este singurul dispozitiv inteligent dintr-un sistem. Susțin acest fapt prin a spune că fiecare dispozitiv conectat la sistem are o parte care îl controlează, care răspunde de acțiunile dispozitivului propriu zis, și care este inteligentă. Această parte poartă numele de controler și realizează acțiunile menționate mai sus prin intermediul unor registrii de stare și control proprii și specifici.

Pe lângă partea hard tangibilă un dispozitiv are și o parte soft. Această parte soft care este menită să programeze un anumit fel de interacțiune cu acesta și să se asigure că acesta va răspunde în mod dorit, practicând astfel un exercițiu de control asupra dispozitivului direcționat prin controler, adică asupra părții fizice care controlează dispozitivul. Această parte soft este păstrată în nucleul Linuxului adică în kernel, și poartă numele de driver de dispozitiv.

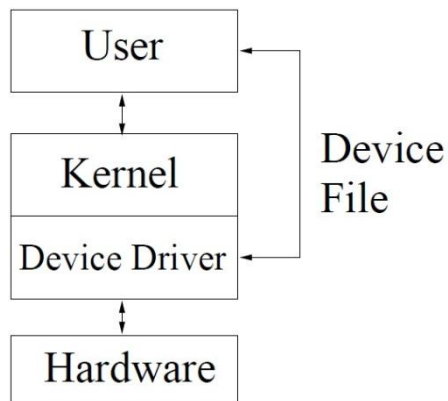


Figura 1: Anatomia unui driver dispozitiv[7]

Dupa cum putem observa în figura de mai sus un driver este alcătuit din mai multe interfețe care reprezintă interacțiunea driverului cu celelalte părți ale sistemului. În primul rând el este făcut să comunice cu partea hard, deci are o interfață concepută pentru acest lucru. O altă interfață este aceea prin care se realizează comunicarea cu utilizatorul, dar cea mai critică interfață este aceea cu ajutorul căreia se realizează comunicarea cu kernelul și fără de care celelalte două interfețe nu ar mai avea sens. În cele ce urmează vom vorbi strict despre interfața dintre kernel și driver.

Interfața kernel-driver

Deoarece kernelul este sursa comună la care trebuie legate toate aceste drivere, ce sunt nevoite să știe anumite particularități ale lui, și din moment ce kernelul este o platforma comună, mereu cunoscută, rezultă că toate interfețele dintre kernel și driver sunt într-un mod sau altul procurate de acestea din urmă.

Acum trebuie să ne putem problema dacă dispozitivul nostru este sau nu suportat de sistem. Ca răspuns la această întrebare Linuxul oferă suport pentru 3 categorii de dispozitive discutate în continuare.

Prima categorie de dispozitive sunt acelea prin care interacțiunea dintre sistem și dispozitiv se realizează prin transmiterea la un moment dat a unui singur caracter de unde și numele de dispozitive caracter. Din descrierea de mai sus survine în mod direct că acestea sunt destinate comunicării de date, cum ar fi modemurile, tastatura. Din punct de vedere al accesării, aceste dispozitive sunt tratate ca fișiere iar pentru a realiza operații asupra lor se folosesc apeluri de sistem.

Un dispozitiv de caracter este disponibil numai după ce driverul său îl face să apară în tabelul dispozitivelor de caracter unde are ca index indentificatorul major. Pe lângă numele driverului acest tabel oferă și un pointer către un block de operații care este presupus a conține adresele rutinelor necesare pentru a opera asupra fișierului care reprezintă driverul. Pentru a fi găsite de către kernel atunci când sunt apelate aceste fișiere dispun de un nod index în sistemul virtual de fișiere care a fost conceput pentru accesarea uniformă a fișierelor. Deoarece acestor noduri le sunt asociate doar operația de deschidere a fișierului atunci când fișierul asociat caracterului este deschis aceasta operație trebuie să acceseze vectorul dispozitivelor de caracter pentru a avea access la zona în care se află rutinele pentru operațiile asociate fișierului de driver respectiv.

A doua categorie de dispozitive sunt dispozitivele bloc, din denumirea lor deducem că transmiterea datelor are loc bloc cu bloc, și deja acest lucru ne trimite cu gândul la dispozitive precum unități hard disk sau flash.

În mod asemănător dispozitivelor caracter, și aceste dispozitive sunt accesate ca fișiere și sunt disponibile atunci când driverul specific înregistrează dispozitivul în vectorul dispozitivelor bloc cu un index ce reprezintă indentificatorul major al sistemului. Deosebirea dintre ele este că în cazul dispozitivelor block care sunt aranjate în clase și care ele însele sunt responsabile pentru înregistrarea în kernel și pentru oferirea unor modalități de operare asupra fișierelor.

O altă deosebire dintre aceste două tipuri de dispozitive este că cele din urmă folosesc un buffer pentru rutinele de intrare ieșire și astfel fiecare driver este nevoit să ofere o interfață pentru acest buffer. Interacția dintre buffer și dispozitiv are loc prin intermediul unor cereri indexate, rutine și structuri de date iar atunci când o cerere este primită, driverul începe să proceseze cererile care sunt memorate într-o structură asociată bufferului și care se blochează automat. După ce termina acest lucru driverul eliberează și golește structura asociată bufferului.

Ultima categorie de dispozitive suportate de linux este cea a dispozitivelor de rețea, care sunt responsabile de transmiterea și receptarea pachetelor de date. Din punctul de vedere al organizării, aceste dispozitive sunt reprezentate ca și în cazul celor de mai sus de o structură proprie și fiecare driver își înregistrează propriul dispozitiv în kernel într-o manieră similară celor de mai sus. Structura dispozitivelor de rețea conține în același timp informații despre dispozitiv precum nume, informații de control, steaguri, informații de protocol, funcții de suport, structuri pentru transmiterea pachetelor și funcții care sunt folosite de protocoalele de comunicații pentru transmiterea datelor folosind dispozitivul respectiv care în mod normal este placa de rețea.

Excepții și întreruperi

Definiția unei întreruperi survine chiar din numele folosit, în mod general poate fi văzut ca un eveniment menit a întrerupe procesorul din execuția curentă. Acest eveniment este în general generat de dispozitive care au și nevoie să desfășoare anumite procese. În orice moment de timp unitatea centrală de prelucrare poate acționa în trei moduri dar cel care are prioritatea cea mai mare este servirea unei întreruperi.

Fiecare întrerupere sau excepție este definită printr-un număr de ordine numit vector. Valorile 0-31 sunt asociate excepțiilor și întreruperilor nemascate, 32-47 sunt asociate întreruperilor mascate iar valoarea 128 este asociată apelurilor de sistem. Fiecare valoare a vectorului este asociată unei entități menite să manipuleze respectivele excepții și întreruperi.

Excepțiile sunt folosite pentru a semnala condiții anormale iar entitatea care le manipulează are o structură care salvează registrele în stiva modului kernel, manipulează excepția folosind funcția C iar apoi iese. Manipulatorii întreruperilor folosesc un alt mod care este compus din salvarea valorii de cererii de întrerupere și executarea cererii de servire asociată.

Concluzie

Pentru a lega cele expuse mai sus le voi rezuma într-o modalitate mai simplistă. Fiecare dispozitiv este compus din 2 părți, o parte fizică reprezentată de dispozitivul propriu zis și o parte programabilă reprezentată de driverul lui. Partea de program are rolul de a ascunde utilizatorului toate procesele care au loc în fundal pentru a asigura desfășurarea unei activități care poate implica mouseul, tastatura, placa

de rețea, imprimanta, hard diskul. Modalitatea prin care aceste dispozitive sunt instalate este prin înregistrarea lor în nucleul sistemului prin diverse structuri de date ale căror câmpuri sunt setate să faciliteze desfășurarea unor acțiuni specifice. Modalitatea practica prin care un dispozitiv realizează ceva este prin a cere realizarea unui serviciu procesorului folosind întreruperi, atunci când apar anomalii în funcționare atunci ele sunt semnalizate prin excepții care împreună cu întreruperile sunt rezolvate cu ajutorul unor entități concepute pentru a le manipula și care dispun de algoritmi specifici. Toate aceste procese nu se pot desfășura dacă nu există o interfață care să le suporte, aceasta interfață este interfața dintre driver și nucleu prin intermediul căreia utilizatorul obișnuit reușește să interacționeze cu dispozitivul fizic. Interfața driver nucleu este chiar zona de separare dintre partea de program și partea fizică și poate fi asemuită unui translator menit să realizeze comunicarea dintre 2 oameni ce vorbesc limbi diferite.

Bibliografie

1. http://www.freesoftwaremagazine.com/articles/drivers_linux
2. http://stanislavs.org/helppc/device_attributes.html
3. <http://www.comedi.org/>
4. http://www.kroah.com/log/linux/stable_api_nonsense.html
5. Kernelul linux: Semnale și întreruperi, Boston University 2001
6. Întreruperile linux: Concepte de bază, Mika J. Järvenpää
7. Introducere în dispozitivele driver linux, Muli Ben-Yehuda
8. Kernelul Linux, David A. Rusling
9. Dispozitive alocate linux, versiunea 2.6+
10. Evenimentul de programare linux Brian W. Kernighan, Rob Pike
11. Sistemul virtual de fișiere al nucleului linux