**GESTIUNEA DE INTRĂRI ŞI IEŞIRI**

**Concepte fundamentale: gestiune de întreruperi, device drivere, gestiune din sistemul de operare, gestiune de către utilizatori**

Studenți: Burlacu Oana

Ciolovan Alexandru

Muhlbacher Louise

Sandu Cristina

**Cuprins:**

**Concepte fundamentale**

1. Introducere
   1. Sisteme de operare
   2. Tipuri de sisteme de operare
   3. Functionarea sistemelor de operare
   4. Tipuri de echipamente I/E
   5. Gestiuni I/E

*Burlacu Oana*

**Driver Mouse**

1.INTRODUCERE………………………………………………………….

2.CONECTAREA HARDWARE INTRE MOUSE SI CALCULATOR…..

3.Moduri de conectare…………………………………………….

3.A. Conectarea prin cablu…………………………………………………..

3.B. Conectarea fără cablu (comunicaţie în infraroşu, radio, bluetooth)…....

3.C. Conectarea direct la tastatură…………………………………………..

3.D. Conectarea direct la magistrală………………………………………...

2.CURSORUL MOUSE-ULUI……………………………………………..

2.1.Cursorul in modul text…………………………………………………..

2.2.Cursorul in modul grafic……………………………………..………….

3.BUTOANELE MOUSE-ULUI……………………………………..….….

4.UNITATEA DE DISTANTA……………………………………..……….

5.PROGRAMAREA MOUSE-ULUI………………………………………..

6.CATEVA FUNCTII ALE INTRERUPERII INT 33h……………………..

7.ECRANUL VIRTUAL………………………………………………………

8.ALTERNATIVE LA TRADITIE……………………………………………

9.CONCLUZII…………………………………………………………………

10.Bibliografie………………………………………………………

*Ciolovan Alexandru*

**GESTIUNEA DE CĂTRE UTILIZATORI**

1. Tipuri de interfețe de intrare/ieşire
2. Modalităţi de transfer de intare/ieşire

3.1. Transferul de I/O prin program

3.2. Transferul I/O prin întreruperi

3.3. Transferul prin acces direct la memorie(DMA)

1. Magistrala de intrare/iesire

* ISA (Industry Strandard Arhitecture)
* MCA (Micro Channel Arhitecture)
* EISA (Extended ISA)
* VL-Bus (VESA Local Bus)
* PCI (Peripheral Component Interconnect)
* AGP

1. Controlere specializate
2. Drivere de dispozitiv
3. Definirea canalelor I/O
4. Concluzii
5. Bibliografie

*Sandu Cristina Maria*

**GESTIUNE DIN SISTEMUL DE OPERARE**

1.Introducere

2. Gestionarea proceselor

2.1. Stările unui proces

2.2.Sincronizarea execuţiei proceselor

Mecanisme de sincronizare a proceselor

Mecanismele de sincronizare

Mecanismele de sincronizare directă,

*Muhlbacher Louise*

Burlacu Oana

* 1. **Sisteme de operare**

De obicei un sistem de operare este ca un fel de concept familiar al tuturor utilizatorilor de calculatoare personale.Acestea in ziua de astazi apar in implementarea de functionalitati ale multor dispozitive: telefoane mobile,tablete etc.Acesta mai poate fi definit ca un soft care imparte si administreaza resursele unui sistem de calcul si are o interfata mai usor de folosit pentru a folosi puterea de procesare a sistemului hardware intr-un mod organizat.

Un sistem de operare are ca scop sa asigure flexibilitatea si si stabilitattea functionarii unui calculator.In general,acestea permit rularea a mai multor aplicatii,intercatiunea a mai multor utilizatori si modificarea strategiilor de raspuns ale calculatorului la o anumita problema.

Intr-un sistem de operare,cea mai importanta componenta este *nucleul* sau *kernetul*. Acestea au rolul de a gestiona resursele fizice si de a oferi o interfata comuna a aplicatiilor. Nucleul nu este o aplicatie, ci un mediu in cadrul caruia pot rula alte aplicatii.Programele de baza sunt acele aplicatii fundamentale ale unui sistem de operare care permit interactiunea dintre nucleu si sistem fizic.

**Functiile unui sistem de operare**

* *Administrarea resurselor hardware ale sistemului* (accesul la procesor,hard-disk,memorie,comunicarea in retea,la alte dispozitive I/E)
* *Controlul asupra calculatorului* (sistemele de operare trebuie sa asigure accesul la resurse)
* *Transparenta fata de modul de functionare a hardware-ului* (sistemele de operare trebuie sa ofere un set limitat de informatii si functii de lucru cu hardwareul)
* *Asigurarea securitatii si integritatii sistemului de calcul*(Sistemele de operare va controla accesul utilizatorilor si proceselor la resursele hardware ale sistemului si previne executia de instructiuni invalide)
* *Portabilitatea* (Deoarece sistemele de operare interactioneaza cu arhitecturi hardware si cu periferice foarte diverse,el trebuie sa faciliteze accesul aplicatiilor la acestea,oferind o interfata unica cu utilizatorul,indiferent de diferentele hardware dintre arhitecturi)

Pentru a comunica cu dispozitivele periferice, sistemele de operare folosesc *drivere.* Driverele sunt niste programe care traduc comenzile sistemelor de operare in comenzi inteliginile echipamentelor,precum si iesirea acestora in mesaje accesibile sistemului.

* 1. **Tipuri de sisteme de operare**

Sisteme de operare de pe smart carduri servesc functii reduse si administreaza resurse foarte limitate.

Sisteme de operare embedded-sunt incluse in dispositive precum televizoarele, telefoanele mobile avand constrangeri specifice functionalitatii lor.

Sisteme de operare in timp real (RTOS-Real Time Operating System)- sunt folosite in echipamente stiintifice, industriale si de precizie. Scopul acestora este de a reduce utilizarea echipamentelor si sa asigure predictibilitatea functionarii lor.

RTOS:

* Sisteme de timp real hard
* Sisteme de timp real soft

Sisteme de operare modern:

* Sisteme multi-tasking
* Sisteme multi-user

Sisteme de operare specializate:

* Pentru situatiile de conectare a calculatoarelor in retea
* Pentru aplicatiile care ruleaza pe un server: Microsoft Windows,Unix,Apple Mac OS X

Sisteme de operare pentru arhitecturi multiprocesor

* 1. **Functionarea sistemelor de operare**

Sistemele de operare au o serie de responsabilitati. Printre acestea putem numi:

* Administrarea procesorului
* Administrarea memoriei
* Administrarea echipamentelor si a perifericelor
* Administrarea sistemelor de stocare a datelor
* Interfata cu aplicatiile si interfata cu utilizatorii
* Organizarea informatiei in sisteme de fisiere

Administrarea perifericelor

Dupa cum am scris si mai sus, administrarea perifericelor si a echipamentelor se realizeaza prin intermediul driverelor. Aceste programe sunt autonome de sistemele de operare, de foarte multe ori sunt furnizate de producator echipamentelor si instalate separat. Acestea permit comunicarea dintre sistemele de operare cu echipamentele noi fara a fi modificate.Comunicarea cu mediul este intermediata de spatii tampon, in care datele de intrare acumulate urmand sa fie transmise procesorului in ritmul accesibil acestuia,evitand congestia atunci cand apare o aglomerare datorita multitudinii de aplicatii.

* 1. **Tipuri de echipamente I/E**



**Exemplu de microcomputer(H.Cucu:Arhitectura microrocesoarelor,2014)**

Managementul dispozitivelor I/O este o parte foarte importantă a sistemului de operare – importanță datorată și diversității subsistemelor I/O dedicate pentru funcționarea acestuia. (De luat în considerare gama de dispozitive de pe un computer modern, de la mouse, tastaturi, unități de disc, adaptoare de afișare, dispozitive USB, conexiuni la rețea, audio I/O, imprimante, dispozitive speciale pentru persoanele cu handicap, și multe periferice cu destinație specială.)

Subsistemele I/O trebuie să realizeze un compromis între cele doua tendinţe:

* inerţia față de interfețele standard pentru o gama largă de dispozitive, ceea ce face mai ușor adăugarea de dispozitive recent dezvoltate pentru sistemele existente;
* dezvoltarea de noi tipuri de dispozitive, pentru care interfețele standard existente nu sunt întotdeauna ușor de aplicat.

***Driverele de dispozitiv*** sunt module care pot fi conectate la un sistem de operare având ca scop interfațarea cu un anumit dispozitiv sau cu o categorie de dispozitive similare.

Funcţiile şi cerinţele principale ale unui modul de I/E se încadrează în următoarele categorii:

* Control şi sincronizare;
* Comunicaţia cu UCP;
* Comunicaţia cu echipamentele externe;
* Bufferarea datelor;
* Detecţia erorilor.

Dispozitivele I/O pot fi clasificate astfel: dispozitive de stocare, dispozitive de comunicații, dispozitive interfață utilizator, și altele.

Tipurile de echipamente putem sa le clasificam in mai multe moduri.Una din clasificari poate fi dupa utilizarea acestora.

In acest caz exista 3 tipuri:

1. Pentru transportul datelor intre utilizator si procesor: echipamente de achizitie si prezentare a datelor
2. Pentru transferul datelor intre mai multe procesoare: echipamente de retea
3. Pentru memorarea informatiilor, ca parte a ierarhiei de memorii a procesorului:echipamente de memorare

In LINUX,putem afirma ca dispozitivele se impart in 2 categorii:

1. Dispozitive caracter
2. Dispozitive bloc

Toate dispozitivele I/O sunt clasificate ca fiind dispozitive (prime) block sau caracter. Dispozitivul special block determină I/O să fie tamponat în piese mari. Caracterul dispozitiv (brut) determină I/O să producă un caracter (octet), la un moment dat. Unele dispozitive, cum ar fi discurile și benzile, pot fi atât block cât și caracter, și trebuie să aibă înregistrări pentru fiecare mod. Terminale funcționează în modul de caracter. Prima intrare în domeniul indică fie b -> block, sau c -> character.

Diferenta dintre acestea se poate observa datorita literii care apare pe prima coloana in iesirea comenzii ls-1,aceasta coloana indica tipul fisierului.Fisierele care au tipul C indica dispozitivele de tip caracter,iar cele care au tipul B indica dispozitivele de tip bloc.

Unele dintre dispozitive suntgenerate de kernel si au componente aparte.Aceste se numesc pseudo-dispozitive.

*Dispozitivele caracter* corespund dispozitivelor care transmit cate un caracter odata. Acestea trimit sau primesc un sir de caractere(stream) fara nicio organizare a informatiei pe adrese,de aceea nu se poate accesa o anumita portiune.

*Dispozitivele bloc* corespund dispozitivelor care lucreaza cu datele la nivel de bloc(citesc,scriu,sterg blocuri de date). Acestea memoreaza informatia in blocuri de dimensiuni fixe(512 la 32,768byte) fiecare cu adresa proprie.Caracteristica fundamentala a acestora este ca informatia poate fi citita doar din blocurile care intereseaza,desigur pentru un cost de timp(ex: hard-disk).

Dispozitivele care functioneaza astfel sunt cele de stocare:

* Fizice - unitati CD-ROM/DVD-ROM, hard-diskuri
* Virtuale – dispozitive loop
* Asociate zonelor de memorie

Dispozitivele de tip bloc sunt mult mai performante decat cele de tip caracter,deoarece viteza celor de tip bloc este mult mai mare decat celelalte.Importanta acesteia este foarte mare,pentru ca LINUX trateaza diferit cele doua tipuri de dispozitive.Pe cand ,Windows, prin intermediul I/E managerul ofera o interfata unificata pentru tratarea dispozitivelor.

Lucrul cu dispozitivele de tip caracter e mult mai usor fata de celelalte.Acestea au o singura pozitie curenta,intimp ce cele de tip bloc trebuie sa se poata misca la orice pozitie din dispozitiv pentru asigurarea accesului aleator la date.Pentru a simplifica lucrurile cu dispozitivele de tip bloc, nucleul LINUX pune la dispozitie un intreg subsistem denumit *subsistemul block I/O (block layer).*

*Pseudo-dispozitivele(ceasurile)* sunt o serie de dispozitive care nu corespund unei componente hardware.Acestea pot fi echivalente unor dispozitive virtuale care implementeaza comportamentul dispozitivelor normale de tip caracter.

Dispozitivele se pot conecta la computer prin intermediul ***porturilor*** , de exemplu, pe portul serial sau paralel.

Un set comun de conexiuni care conecteaza mai multe dispozitive este numit ***bus.***

Figura 1 de mai jos ilustrează trei dintre cele patru tipuri de bus-uri care se găsesc într-un PC modern:

* + - ***PCI***  interfaţeaza dispozitive de mare viteza cu subsistemul de memorie (și CPU.);
    - ***Bus-ul de expansiune*** conectează dispozitive lente;
    - ***Bus-ul SCSI*** conectează un număr de dispozitive SCSI la un controler SCSI comun;
    - Bus-ul ***daisy-chain,***  este atunci când un șir de dispozitive este conectat între ele ca mărgelele pe lanț, și doar unul dintre dispozitivele este conectat direct la gazdă.

Sistemele de operare trebuie sa ia in considerare rata de transfer a informatiei a dispozitivelor de intrare/iesire.

Dispozitivele care sunt folosite pentru a introduce datele și programele din computer sunt cunoscute sub numele de "Dispozitive de intrare". Un dispozitiv de intrare are funcţia de a citi datele și de a le converti într-o formă pe care un calculator poate să le utilizeze.

Dispozitivele de ieșire transformă produsul final al prelucrării mașinii într-o formă utilizabilă de către om. Acesta oferă comunicarea computerului cu utilizatorul. Unele dintre dispozitivele I / O sunt explicate mai jos:

**Tastatura:** este utilizată în etapa de introducere al unui grup de informaţii în calculator. Tastatura este dispozitivul de intrare cel mai frecvent utilizat în zilele noastre. Datele și instrucțiunile sunt introduse prin apăsarea tastelor de pe tastatură. Mesajul dactilografiat ajunge în memoria calculatorului. Tastatura este conectată la un calculator printr-un cablu. În afară de taste alfabetice și numerice, conţine şi alte taste pentru efectuarea de funcții diferite.

**Mouse-ul:** Este un dispozitiv de indicare. Mouse-ul este mișcat pe mouse pad, care, la rândul său, controlează mișcarea cursorului pe ecran. Putem face clic, dublu clic sau să glisăm cu ajutorul mouse-ul.

Mouse-ul optic nu utilizează o minge de rulare, dar utilizează în schimb o lumina şi un senzor optic de mici dimensiuni pentru a detecta mișcarea mouse-ului prin urmărirea unei imagini mici a suprafeței biroului. Mouse-urile optice evită problema de murdarire a mingei mouse-ul, care cere o curaţare regulata pentru a se rostogoli fara probleme .

Un mouse fără fir sau wireless comunică cu computerul prin intermediul undelor radio (de multe ori folosind tehnologia Bluetooth și protocol), astfel încât nu este nevoie de un cablu (dar astfel de mouse-uri au nevoie de baterii interne).

**Scanner-ul:** Scanerele sunt folosite pentru a introduce informaţii direct în în memoria calculatoarelor. Acest dispozitiv funcționează ca un aparat xerox. Scanner-ul transformă orice tip de informaţii tipărite sau scrise, inclusiv fotografii în impulsuri digitale, care pot fi manipulate de către calculator.

**Touchscreen.** Unele calculatoare, în special PDA-uri mici de mână, au ecrane cu atingere sensibila. Utilizatorul poate face alegeri și imagini apăsând butonul de pe ecran. Puteți utiliza de multe ori un stylus, care arată ca un stilou, pentru a "scrie" pe suprafața de un ecran tactil mic.

**Microfon.** Un microfon poate fi atașat la un computer pentru a înregistra un sunet (de obicei, printr-o intrare a placii de sunet sau un circuit construit pe placa de bază). Sunetul este digitizat, şi transformat în numere care reprezintă sunetul analogic inițial și stocate în calculator pentru prelucrare și redare mai târziu.

**Bar Code Reader**: Acest dispozitiv citeste coduri de bare si le converteste în impulsuri electrice pentru a fi procesate de către un calculator. Un cod de bare nu este nimic altceva decât date codificate în formă de bare de lumină și întuneric.

**Monitor cu ecran plat.** Un ecran plat foloseste , de obicei, un ecran LCD (Liquid Crystal Display ) pentru a afișa date de ieșire de la calculator . Ecranul LCD este format din mai multe straturi subțiri care polarizează trecerea luminii prin ele . Polarizarea pe singur strat , care conține molecule lungi subtiri numite cristale lichide , poate fi controlat electronic la fiecare pixel, prin blocarea cantități variabile de lumina pentru a face un pixel de intensitate mai deschisă sau mai închisă . Alte tipuri de tehnologie cu ecran plat exista ( cum ar fi ecranele cu plasmă ), dar LCD-urile sunt cele mai frecvent utilizate pentru calculatoare , mai ales laptop-uri.

**Imprimantă cu jet de cerneală.** Pentru tipărirea hârtiei, avem nevoie de o imprimantă atașată la computer (sau într-o rețea). Cel mai comun tip de imprimantă pentru sistemele de acasă este imprimanta color cu jet de cerneală. Aceste imprimante formează imaginea pe pagină prin pulverizare în picături mici de cerneală de la capul de imprimare. Imprimanta are nevoie de mai multe culori de cerneală (cyan, galben, magenta și negru) pentru a face imagini color. Unele imprimante foto de calitate cu jet de cerneală au mai multe culori de cerneală.

Imprimante cu jet de cerneală sunt ieftine, dar costul consumabilelor (cartușe de cerneală și hârtie specială) le face costisitoare pentru a opera pe termen lung în mai multe scopuri.

**Imprimantă cu laser.** O imprimantă cu laser produce imagini de bună calitate cu aceeași tehnologie pe care o folosesc şi fotocopiatoarele. Un tambur acoperit cu material fotosensibil este încărcat, apoi o imagine este scrisă pe el cu un laser (sau LED-uri), care face ca pe aceste zone să apară pierderi sarcină. Toba cu role trece apoi prin toner (particule de plastic mici de pigment), care sunt atrase de zonele încărcate ale tamburului. Tonerul este apoi depus pe hârtie, și apoi topit în hârtia cu căldură.

Cele mai multe imprimante sunt cu laser monocrom (doar o singura culoare, de obicei negru), dar imprimante laser mai scumpe cu cartușe de toner de culoare multiple poate produce rezultate color.

**Ieşirea audio.** Calculatoare conţin, de asemenea, ieșire audio, variind de la beep-uri simple de alertare a utilizatorului, până la efecte de sunet joc impresionante, şi la muzică de calitate. Circuitele pentru a produce un sunet pot fi incluse pe placa de baza, dar ieșirea audio de înaltă calitate de la un PC necesită, de obicei, o placă de sunet într-unul dintre sloturile de expansiune, conectat la un set de boxe externe de bună calitate sau căști.

**Hard disk.** O unitate de hard disk conține discuri din metal ce sunt acoperite cu un oxid metalic, care poate fi magnetizat. Un cap mic de citire/scriere electromagnetic aflat la capătul unui braț caută şi magnetizează pete mici de pe disc pentru a stoca date. Petele magnetice magnetizate într-o singură direcție reprezintă unu; petele magnetizate în direcție opusă reprezintă un zero Cu același cap electromagnetic putem citi mai târziu datele de pe disc.

Hard disk-urile sunt evaluat in funcţie capacitatea lor de stocare, de obicei, zeci sau sute de gigabytes. Ele sunt, de asemenea, evaluate în funcţir de cât de repede se învart discurile (în rpm, rotatii pe minut), care este de obicei mii de rotații pe minut. Un alt mod de a evalua un hard disk este timp mediu de acces (măsurat în milisecunde, ms), care spune, în medie, cât timp i-ar lua unităţii pentru a prelua orice bit de date de pe disc.

**USB flash**. O unitate USB flash este un dispozitiv portabil de memorie, care se conectează la un port USB al computerului. Au mai multe denumiri cum ar fi unitatea-cheie, unitate de buzunar, unitate degetul mare, pen drive. Au înlocuit dischete și discuri Zip la UNM-LA ca mijloacele noastre preferate pentru a transporta fișiere. Se pot folosi pe Mac-uri și PC-uri.

Aceste unități mici stocheayă a datele pe microcipuri de memorie flash (un fel de EEPROM). Memoriile flash pot fi șterse și re-scrise de un număr limitat de ori (de obicei, mai multe mii de ori). Unele unități au un comutator de protecție la scriere.

Capacitatea de stocare variază, de la 16 MB la peste un gigabyte.

**CD-ROM-ul.** Un CD-ROM (Compact Disc Read Only Memory) este un mediu de stocare optic care poate stoca aproximativ 670MB. "Optic" înseamnă că lumina este utilizată pentru a citi datele de pe disc (acesta nu este un mediu magnetic). CD-ROM-urile sunt foarte ieftine pentru a produce în cantități mari, astfel încât cele mai multe cazuri software-ul este distribuit pe CD-ROM-uri.

Datele sunt stocate pe un CD-ROM-ul ca gropi mici din plastic pe un strat interior, care este apoi aluminizată și acoperit cu un alt strat transparent. Un fascicul laser în unitatea CD-ROM este proiectat pe disc și pe secvențele de gropi și peste gropi reflectivitatea este diferită,apoi este transformată în unu și zero.

CD-ROM-urile sunt evaluate prin viteza, cum ar fi 32x, ceea ce înseamnă de 32 ori mai rapid decât primele unități CD-ROM.



**Exemple de dispozitive periferice** [**(http://goo.gl/YhNT7q)**](file:///G:\poli%20faculatate\An%203\sem2\SO\tema\(http:\goo.gl\YhNT7q))

* 1. **Gestiuni I/E**

Toate computerele au componente hardware care citesc date de intrare,afiseaza sau pentru pastrarea rezultatelor procesarii. Aceste dispozitive pot fi imprimante,tastaturi,monitoare si multe altele.Toate acestea au un sistem pentru a le utiliza, astfel ca sistemele au un subsistem de intrare/iesire specializat in acest sens.

Dupa cum stim de la arhitectura microprocesoarelor,spatiul de adrese presupune o serie de adrese,fiecare corespunzand unui registru fizic sau virtual,un I/O device,un sector de disk sau alta entitate fizica sau logica.O adresa pointeaza spre locul in care se gaseste informatia cautata.

Gestiunea de intrari/iesire se face prin doua metode complementare:

1. *Memory-mapped I/O (MMIO) (memoria mapata de intrare/iesire*)- foloseste aceleasi registre de adresare si aceeasi magistrala pentru memorie si dispozitive I/E. Adresele si registrii dispozitivelor I/O sunt asociate cu valoarea adresei.Atunci cand o adresa este acceptata de CPU,se poate referi la o parte a memoriei RAM,dar se poate referi si la memoria dispozitivelor I/O.Deci,instructiunile CPU folosite pentru a accesa memoria pot fi utlizate si pentru accesarea dispozitivului.Fiecare dispozitiv I/O monitorizeaza adress buss al CPU-lui si raspunde oricarui acces al CPUlui de la o adresa atribuita dispozitivului,conectand data bus in registrul hardului dispozitivului dorit.Ca sa acomodam dispozitivele I/O,locatiile de memorie folosite de CPU trebuie sa fie ocupate de I/O si nu trebuie sa fie disponibile pentru memoria fizica.Rezervarea poate fi temporara sau permanenta pentru aceste locatii de memorie.
   * În acest caz, o anumită parte a spațiului de adrese a procesorului este mapata la dispozitiv, și comunicațiile au loc prin citirea și scrierea directă la/de la zonele respective de memorie.
   * Această tehnica este potrivită pentru dispozitivele care tranferă rapid cantități mari de date, cum ar fi cardurile de grafica.
   * Aceasta tehnica poate fi folosită în același timp sau în combinație cu registrele tradiționale. De exemplu, plăci grafice încă mai folosesc registrele de informații de control, cum ar fi setarea modului video.

**Polling** este un mijloc simplu de protocol de comunicare cu dispozitivele :

Gazda verifică în mod repetat ***bitul ocupat*** de pe dispozitiv, până când devine 0.

* + Gazda scrie un octet de date în registrul de date-out, și setează ***bitul de scriere*** în registrul de comandă
  + Gazda setează ***bitul*** ready ***de comandă*** în registrul de comandă a dispozitivului.
  + Când controller-ul dispozitivului vede setat bitul set-comand ready, se setează bitul busy.
  + Apoi, controlerul dispozitivului citește registrul de comandă, vede setat bit-ul de scriere, citește octetul de date din registrul de date-out, și trimite octetul de date.
  + Controler-ul dispozitivului șterge ***bitul de eroare*** în registrul de stare, bit-ul de comandă ready, iar în final se șterge bitul busy, semnalizand astfel încheierea operațiunii.

**Polling** poate fi foarte rapid și eficient, în cazul în care atât dispozitivul și controlerul sunt rapide și în cazul în care există date semnificative pentru a transfera. Polling-ul devine ineficient, cu toate acestea, în cazul în care gazda trebuie să aștepte o perioadă lungă de timp în bucla de așteptare pentru dispozitiv, sau în cazul în care cantitătile de date ce urmează a fi transferate sunt reduse.



**Tehnica polling(H.Cucu:Arhitectura microrocesoarelor,2014)**

1. *Port-mapped I/O (PMIO)-* foloseste un set de registre speciale pentru a accesa dispozitivele I/O .Acestea au o mapare speciala rezervata pentru ele si uneori chiar o intreaga magistrala sau instructiuni. Avantajul acestui mecanism este la microprocesoarele cu spatiu de adresare limitat, asta insemnand ca duce la o gestionare mai buna a memoriei principale,dar se poate spune ca duce un plus asupra usurintei cu care poate fi scris/citit un cod in limbaj de assamblare datorita instructiunilor dedicate.

Aceasta modalitate de a comunica cu dispozitivele este prin ***registre*** asociate cu fiecare port. Registrele pot fi de unu la patru octeți în mărime, și pot include în mod tipic următoarele:

***Registrul de date-in*** este citit de către gazdă pentru a obține informații de la dispozitiv.

***Registrul de date-out*** este scris de gazda pentru a trimite spre ieșire.

***Registrul de stare*** are biți cititi de către gazdă pentru verificarea stării dispozitivului, cum ar fi inactiv, gata de intrare, ocupat, eroare, tranzacție completa, etc

***Registrul de control*** are biți scrisi de gazda pentru a emite comenzi sau pentru a modifica setările dispozitivului, cum ar fi verificarea parității, lungime cuvânt, sau full/ semi-duplex.

Conceptele fundamentale pe care se bazeaza acest subsistem sunt

- gestiune de intreruperi (interrupt handlers)

- driveri de dispozitive (device drivers)

- gestiune de catre sistemul de operare (kernel level mode)

- gestiune de catre utilizator(user level mode)

- principiile de intrare/iesire hardware (input/output hardware)

Bibliografie:

* *<http://users.utcluj.ro/~baruch/book_ac/AC-Unitatea-IE.pdf>*
* *<http://en.wikipedia.org/wiki/Memory-mapped_I/O>*
* *„Introducere in sisteme de operare”-R Rughinis,R Diaconescu,Ge Milescu,M Bardac,Ed Printech*
* *<http://www.rasfoiesc.com/educatie/informatica/calculatoare/Gestiunea-sistemului-de-IO58.php>*
* [*http://electronica07.curs.ncit.pub.ro/course/category.php?id=2*](http://electronica07.curs.ncit.pub.ro/course/category.php?id=2)
* *<http://cristis.hi2.ro/computer/hardgen.php?no=1>*
* *<http://vega.unitbv.ro/~romanca/AOC/Cap7-AOC-IntrareIesire.pdf>*
* *<http://users.utcluj.ro/~baruch/book_ac/AC-Unitatea-IE.pdf>*
* *<http://www.referatele.com/referate/informatica/online7/DISPOZITIVE-DE-INTRARE-IESIRE-referatele-com.php>*

Ciolovan Alexandru

**Device Drivere-Driver Mouse**

1.INTRODUCERE

Pentru utilizarea mouse-ului pe un calculator,sunt necesare trei elemente de baza:calculatorul,mouse-ul si driverul de mouse.Acesta din urma este un program care preia datele oferite de mouse si le transforma in deplasarea pe ecran a unui cursor.Acest program pune la dispozitia instalatorului de mouse diverse functii care permit setarea formei cursorului,a vitezei lui de deplasare,verificarea butoanelor etc.

Exista o intrerupere software care permite accesul la rutinele de programara a mouse-ului,si aceasta este interfata de mouse Microsoft INT 33h.Functiile oferite de aceasta interfata sunt suportate in mare masura si de asa-zisele drivere compatibile Microsoft.

Pentru instalare se poate apela intreruperea INT 33h prin intermediul intreruperii INT 86h,care este interfata generala cu intreruperile software.Vom exemplifica in limbajul C cum se poate instala mouse-ul cu ajutorul unui limbaj de nivel inalt:

# include <dos.h>

...

union REGS reg;

...

main()

{

reg.x.ax=0x01;

int86(0x33,&reg,&reg);

...

}

2.CONECTAREA HARDWARE INTRE MOUSE SI CALCULATOR

Mouse-ul se conecteaza cu calculatorul printr-o interfata V24.La fiecare 100 ms, calculatorul scaneaza iesirea mouse-ului,care transmite trei octeti de date.Primul octet reprezinta deplasarea pe directia x in cele 100 ms de la citirea anterioara,al doilea octet contine deplasarea pe directia y in acelasi interval de timp,iar ultimul octet contine starea logica a butoanelor mouse-ului.Avand la dispozitie aceste date,driver-ul de mouse are sarcina sa le analizeze si sa le transforme in pozitii absolute ale cursorului pe ecran.

Driver-ul de mouse este un program care preia datele oferite de mouse si le transforma in deplasarea pe ecran a unui cursor.Acest program pune la dispozitia instalatorului de mouse diverse functii care permit setarea formei cursorului,a vitezei lui de deplasare,verificarea butoanelor,etc.

3.Moduri de conectare

Legătura cu sistemul de calcul se poate realiza:

A-prin cablu;

B-fără cablu (comunicaţie în infraroşu sau prin unde radio);

C-direct la tastatură;

D-direct la magistrală.

A. Conectarea prin cablu

Mouse-ul se conectează la calculator prin unul din porturile seriale COM1 şi COM2, pe portul PS/2 (direct pe placa de bază) sau, mai nou, pe portulUSB. Informaţiile sunt transmise de către mouse la fiecare 500-600 ms şi sunt recepţionate de calculator prin interfaţa V24. Aceste informaţii conţin 3 octeţi. Primul octet reprezintă deplasarea pe coordonata x în ultimele 100 ms citite, al doilea reprezintă deplasarea pe coordonata y în acelaşi interval, iar al treilea, starea logică a butoanelor. Driver-ul analizează starea mouse-ului şi transformă informaţiile referitoare la deplasările relative în poziţii absolute pe ecran şi afişează cursorul.

La fel ca la alte dispozitive seriale, conectorul de la capătul cablului este de tip „tată”, fie cu 9, fie cu 25 de pini. Deşi pentru comunicaţia dintre mouse şi driver-ul de dispozitiv sunt folosiţi numai câţiva pini ai conectorului DB-9 sau DB-25, conectorul prezintă toţi cei 9 sau 25 de pini. Deoarece la majoritatea calculatoarelor PC sunt cel puţin două porturi seriale, mouse-ul serial poate fi conectat atât la COM1, cât şi la COM2. La iniţializarea sistemului, driver-ul de dispozitiv examinează porturile, pentru a determina la care dintre ele este legat mouse-ul. Cum un mouse nu se conectează direct la sistem, nu el este cel care îi foloseşte resursele, ci portul serial corespunzător lui.

Exemplu. Pentru un mouse conectat la COM2, acesta va folosi întreruperea IRQ 3 şi adresele de port I/O 2F8H-2FFH.

B. Conectarea fără cablu (comunicaţie în infraroşu, radio, bluetooth)

Comunicaţia se desfăşoară în infraroşu sau prin unde radio sau bluetooth (mai nou), de aceea acest tip de mouse conţine două dispozitive: mouse-ul propriu zis, împreună cu dispozitivul de emisie în infraroşu/radio, şi dispozitivul de recepţie, cuplat prin fir la calculator. Deplasarea mouse-ului este codată şi transferată prin radiaţie circuitului de decodare, care, la rândul său, trimite semnalul decodat calculatorului, semnalul fiind standard.

C. Conectarea direct la tastatură

Al treilea tip de mouse este şi cel mai nou. De aceea nici forma nu mai este standard, numele nu mai este acelaşi, deşi funcţia este identică: poate fi întâlnit sub denumirile de trackball sau touchpad.

Trackball-ul este un dispozitiv înrudit cu mouse-ul, diferenţa constând în faptul că mouse-ul se mişcă pe un suport pe care acţionează bila, iar la trackball bila se roteşte direct cu mâna. Are, de asemenea, două sau trei butoane. Trackball-ul poate fi integrat în tastatură, sau poate fi construit ca un mouse obişnuit, cu diferenţa că utilizatorul manevrează direct bila, iar mişcările bilei sunt traduse în mişcări ale cursorului pe ecran.



Fig. 3.4. Trackball.

Touchpad-ul este un mic ecran senzitiv, care „simte” degetul. Astfel, cursorul de pe ecran devine prelungirea degetului utilizatorului, acesta fiind un mod de utilizare natural al calculatorului. De asemenea, atingerea mai puternică a ecranului senzitiv are rol de clic, iar atingerea dublă are rol de dublu clic. Şi touchpad-ul este prevăzut cu butoane, de obicei, două. Acesta este folosit în special la notebook-uri, deoarece prezintă avantajul de a fi integrat direct în laptop, iar utilizatorul nu mai trebuie să aibă încă un dispozitiv suplimentar pentru a folosi computerul la întreaga sa capacitate. Unii constructori de notebook-uri (în special, IBM) preferă integrarea unui mouse de tip mini-stick în loc de touchpad.

Figura 5.5 prezintă atât touchpad-ul cât şi stick-ul (dreapta sus).



Fig. 3.5. Touchpad.

D. Conectarea direct la magistrală

De regulă, un mouse pentru magistrală este folosit în sistemele care nu dispun de un port pe placa de bază sau de porturi seriale libere. Numele de „mouse pentru magistrală” derivă din faptul că mouse-ul necesită o placă specială pentru interfaţa de magistrală, care ocupă un conector din calculator (slot ISA) şi comunică cu driver-ul de dispozitiv prin magistrala principală a plăcii de bază. Un mouse de magistrală poate fi periculos, pentru că foloseşte un conector mini-DIN, exact ca mouse-ul de tip PS/2, deşi sunt total incompatibile. Această incompatibilitate poate duce la distrugerea plăcii de bază.

Plăcile şi adaptoarele pentru mouse-ul de magistrală dau, de regulă, posibilitatea alegerii întreruperilor şi stabilirii adreselor portului I/O, dar selecţia IRQ este limitată la întreruperile pe 8 biţi. De obicei, aceasta înseamnă că trebuie selectat IRQ 5 în majoritatea sistemelor care au două porturi seriale, deoarece toate celelalte întreruperi pe 8 biţi vor fi folosite. Dacă, în plus, se doreşte utilizarea unei alte plăci pe 8 biţi, ca, de exemplu, o placă de sunet (care are nevoie de o întrerupere), aceasta nu va putea fi folosită de sistem fără să apară conflicte. Denumirea brevetată a conexiunii de mouse pentru magistrală este aceea de inport mouse (Microsoft).

2.CURSORUL MOUSE-ULUI

Sunt doua tipuri de baza de cursor:cursorul in modul text si cursorul in modul grafic.La un moment dat,pe ecran se va afisa un singur tip de cursor.

2.1.Cursorul in modul text

Cursorul in modul text dispune la randul lui de doua tipuri individuale:cursorul hardware si cursorul software.

a) Cursorul hardware este cel care apare pe ecran la pornirea calculatorului.Este un cursor clipitor care se deplaseaza de la caracter la caracter,putand lua forma unui bloc sau a unei liniute.El este format dintr-un bloc de 8 pixeli latime pe 16 pixeli inaltime.Un set orizontal de pixeli formeaza o linie de baleiere,acestea fiind cele care determina aparitia cursorului.Ele sunt numerotate in general de la 0 la 7 (sau de la 0 la 11,in functie de monitor),linia din varf avand numarul 0.Daca o linie este aprinsa,in zona respectiva va apare clipirea pe ecran,in caz contrar ea neavand efect.

b) Cursorul software se deplaseaza tot de la caracter la caracter,dar utilizeaza atributele da afisare pentru a influenta modul de aparitie al caracterelor pe ecran.Acest efect este creat prin doua masti a cate 16 biti fiecare,masca ecranului si masca cursorului.Valorile acestor determina noile atribute ale caracterelor,in momentul in care ele sunt acoperite de cursor.Masca ecranului decide care atribute ale caracterului se vor pastra,iar masca cursorului in ce fel se vor modifica atributele pentru a genera cursorul.In primul rand,se realizeaza un si logic intre masca ecranului si bitii ce definesc caracterul de pe ecran,iar intre rezultatul obtinut si masca cursorului se efectuaza un sau excusiv.

2.2.Cursorul in modul grafic

Cursorul in modul grafic reprezinta o forma ce se deplaseaza pe ecran peste imagine si consta intr-un bloc de pixeli.Deplasarea pe ecran afecteaza punctele din spatele cursorului,astfel fiind create forma si culoarea sa.Locatia pe ecran a acestui tip de cursor este legata direct de un pixel din spatele sau.Acest pixel,utilizat de software pentru a determina coordonatele cursorului,se numeste punct de referinta.

Si aici se definesc doua masti pentru crearea cursorului.Masca ecranului determina care pixeli vor da forma cursorului si care vor reprezenta fondul,iar masca cursorului care pixeli vor contribui la culoarea cursorului.In momentul in care se fac modificari in zona ecranului in care se afla cursorul,acesta trebuie ascuns pentru a nu fi readuse pe ecran vechile valori.In primul rand se face un si logic intre masca ecranului si bitii de date care definesc pixelii de sub cursor,apoi intre rezultatul obtinut si masca cursorului se efectueaza un sau exclusiv.

3.BUTOANELE MOUSE-ULUI

Fiecare apasare pe un buton determina incrementarea unui contor (pentru memorarea numarului de apasari si eliberari),acesta fiind pozitionat automat pe 0 in momentul citirii sale sau in cazul unei resetari.Ca urmare,starea butoanelor (apasari,eliberari precum si starea logica apasat sau liber) poate fi obtinuta prin metode software,ea fiind specificata printr-o valoare intreaga.Daca bitul corespunzator unui buton are valoarea 1,acesta este apasat,in caz contrar el fiind liber.

4.UNITATEA DE DISTANTA

Miscarea mouse-ului se exprima in unitati de distanta.Unitatea de masura pentru acesta miscare se numeste rata de deplasare,ea reprezentand cel mai mic increment pe care il poate detecta mouse-ul (aproximativ 1/200 inch).

O data cu deplasarea mouse-ului,driverul stabileste un contor de miscare pe verticala si pe orizontala,utilizat pentru deplasarea cursorului pe ecran cu un anumit numar de caractere sau pixeli.De asemenea,se poate defini senzitivitatea mouse-ului,aceasta reprezentand numarul de rate de deplasare necesare pentru a misca cursorul cu 8 pixeli.

5.PROGRAMAREA MOUSE-ULUI

Primul lucru pe care trebuie să-l știm este cum să-i spun unui mousesa faca ceva. În realitate nu comunicam cu mouse-ul direct, ci prin intermediul driverului furnizat.Se folosesc "Întreruperi" pentru a avea acces la acest driver. Fiecare dispozitiv furnizat de un computer are un port unic, care este o valoare hexazecimală si este proiectat pentru a fi independent in functie de program. Mouse-ul are port de 0x33 atașat la acesta. De asemenea, se face uz de registre de adrese. Acestea sunt, practic, de tip UNIUNEA REGS definite în "dos.h". Se folosesc două registre pentru a comunica cu un device driver,un dispozitiv de intrare și unul de ieșire

Intreruperea software care permite accesul la rutinele de programare ale mouse-ului este interfata de mouse Microsoft INT 33h. Functiile oferite de aceasta sunt suportate in foarte mare masura de drivere-le “ compatibile “ Microsoft .

Pentru apelarea unei functii INT 33h , se procedeaza astfel :

- se incarca numarul functiei in registrul reg AX

- se incarca (daca este necesar ) celelalte date in registrii specificati

- se genereaza INT 33h

Exemplu :

mov ax , 01h /\* afisare cursor \*/

int 33h

Functiile INT 33h pot fi apelate si dintr-un limbaj de nivel inalt , ale carui module pot fi legate cu module scrise in limbaj de asamblare . De exemplu secventa celor doua instructiuni prezentata mai sus se poate scrie astfel in Limbajul C :

# include <dos.h>

union REGS reg ;

main ( )

{- - - - - - - - - - - -

reg . x . ax = 0 x 01 ;

int 86 ( 0 x 33 , &reg , &reg ); /\* int 86 -> interfata generala cu intreruperile \*/

Driverul de mouse permite doua tipuri de baza de cursor in functie de modulul video - text si grafic.

Modulul text pernite doua tipuri de cursor: hardware sau software. Pentru localizarea acestui tip de cursor se folosesc coordonatele caracterului din spatele cursorului.

Cursorul hardware este cel care apare pe ecran in momentul pornirii calculatorului. Este un cursor clipitor care se deplaseaza pe ecran de la un caracter la altul luand forma unui bloc (de 8 pixeli latime si 16 pixeli inaltime) sau liniute.

Cursorul software se deplaseaza tot de la un caracter la altul dar utilizeaza caracterele de afisare pentru a influenta modul de aparitie al caracterelor pe ecran.

Acest efect este creat prin doua masti pe 16 biti: masca ecranului decide ce atribute ale caracterelor se vor pastra si masca cursorului care decide in ce fel se vor modifica atributele pentru a se genera cursorul.

Cele doua masti au urmatoarea descriere:

BITI EFECT

0 - 7 specifica codul caracterului

8 - 10 determina culoarea textului

12 - 14 determina culoarea de fond

15 pozitionat pe 1 determina clipirea cursorului

Crearea cursorului se lucreaza cu datele care definesc caracterul pe ecran. Pentru masca ecranului se realizeaza un SI LOGIC iar pentru masca cursorului se efectueaza un SAU EXCLUSIV.

Cursorul grafic reprezinta o “forma” care se deplaseaza pe ecran peste imabine, constituit dintr-un bloc de puncte ecran. Locatia pe ecran a acestui tip de cursor este legata direct de un punct ecran aflat in spatele cursorului. Acest punct se numeste punct de referinta.

Interactiunea pentru crearea cursorului este definita prin doua tabele a cate 16 biti - masca ecranului determina cate puncte ecran dau forma cursorului si care reprezinta fondul, iar masca cursorului determina care ppuncte ecran contribuie la culoarea cursorului.

Pentru culoarea cursorului se lucreaza cu date din memoria ecran care definesc culorile punctelor ecran.Se realizeaza un SI LOGIC (AND) intre masca ecranului si bitii de date care definesc punctele ecran de sub cursor,iar intre rezultatul obtinut si masca cursorului se efectueaza un SAU EXCLUSIV(XOR),rezultind urmatorul tabel:

Bit masca ecran Bit masca cursor Bit ecran rezultat

0 0 0

0 1 1

1 0 nemodificat

1 1 inversat

Pentru inalta rezolutie blocul de pixeli are dimensiunile 16\*16 pentru rezolutie medie 8 culori 4\*16 ,iar pentru rezolutie medie 4 culori 8\*16.

Fiecare apasare pe un buton determina incrementarea unui contor,acesta fiind pozitionat automat pe zero in momentul citirii sale sau a unei resetari.Starea butoanelor poate fi obtinuta prin software,avand valoare intreaga.Butonul este apasat daca bitul corespunzator lui are valoarea 1.

Bit Corespondenta

0 Buton stanga

1 Buton dreapta

2 Buton mijloc

Unitatea de masura pentru miscarea mouse-ului se numeste rata de deplasare,ea reprezentand cel mai mic increment pe care mouse-ul il poate detecta(1/200 inci).

Se poate defini senzitivitateamouse-ului(a se vedea functia 0fh),aceasta reprezentand numarul de rate de deplasare necesara pentru a misca cursorul cu 8 puncte ecran.

Functia 000fh-Definirea senzitivitatii mouse-ului

Functia permite corelarea deplasarii cursorului pe ecran cu miscarea fizica a mouse-ului.

Pentru apelul functiei ,se folosesc registrii:

AX -incarca cu 000fh;

CX -incarca cu valoarea senzitivitatii pe orizontala;

DX -incarca cu valoarea senzitivitatii pe verticala;

Functia nu returneaza nimic.

Exemplu

#include <stdio.h>

#include<dos.h>

union REGS reg;

main()

{

reg.x.ax=1; /\*AX=1 -afisare cursor\*/

int86( 0x33, &reg, &reg); /\*apel INT 33h\*/

getch();

reg.x.ax=0xf; /\*AX=1 \*/

reg.x.cx=250; /\* deplasare orinzontala\*/

reg.x.dx=250; /\*deplasare verticala\*/

int86(0x33, &reg, &reg); /\*apel INT 33h\*/

getch();

reg.x.cx=8; /\*refacere conditii initiale\*/

reg.x.dx=16;

int86(0x33, &reg, &reg); /\*apel INT 33h\*/

Primul program de controale este pentru a vedea dacă un driver de mouse este încărcat sau nu și că programarea mouse-lui este susținută sau nu. Driverul este un program care detectează prezența mouse-ului și înțelege un semnal care vine de la portul mouse-ului înainte de a traduce aceste semnale în acțiuni relevante.

Program scris in C care detecteaza mouse-ul:

#include <dos.h>

union REGS in, out;

void detect\_mouse ()

{

in.x.ax = 0;

int86 (0X33,&in,&out); //invoke interrupt

if (out.x.ax == 0)

printf ("\nMouse Failed To Initialize");

else

printf ("\nMouse was Succesfully Initialized");

}

int main ()

{

detect\_mouse ();

getch ();

return 0;}

În programul de mai jos, avem o buclă while. Această buclă continuă până când o tasta este lovita. În buclă, am folosit sub-funcția 3 și am invocat întreruperea mouse-ului. Sub-funcția 3 întoarce X->coordonata în registrul cx și Y-> coordonata în registrul dx. Declarațiile printf imprimă x și coordonează y atâta timp cât bucla continua. Rezoluția maximă a ecranului pentru mouse-ul în modul text este de 640x200 și în modul grafic este de 640x480.

#include <dos.h>

#include <graphics.h>

union REGS in, out;

void detect ()

{

while (!kbhit () )

{

int x,y;

in.x.ax = 3;

int86 (0X33,&in,&out);

if (out.x.bx == 1)

{

x = out.x.cx;

y = out.x.dx;

printf ("\nLeft || X - %d Y - %d", x, y);

}

delay (200); // Otherwise due to quick computer response 100s of words will get print

}

}

int main ()

{

detect\_mouse ();

showmouse\_text ();

detect ();

hide\_mouse ();

getch ();

return 0;

}

6.CATEVA FUNCTII ALE INTRERUPERII INT 33h

In cele ce urmeaza vom studia pe larg cateva functii mai importante ale interfetei software INT 33h

6.1.Functia 0000h

Aceasta functie produce resetarea mouse-ului la valorile implicite si permite determinarea starii curente a mouse-ului (hardware si software).Pentru apel,se incarca in registrul ax valoarea 0000h.

6.2.Functia 0001h

Aceasta functie permite afisarea pe ecran a cursorului mouse-ului.Pentru apel,se incarca in registru ax valoarea 0001h.Functia nu returneaza nimic.

6.3.Functia 0002h

Aceasta functie realizeaza ascunderea cursorului mouse-ului.Pentru apel,se incarca in registrul ax valoarea 0002h.Functia nu returneaza nimic.

6.4.Functiile 0003h si 0004h

Functia 0003h permite aflarea coordonatelor ecran ale cursorului mouse-ului si a starii butoanelor.Pentru apel,se incarca in registrul ax valoarea 0003h.Functia returneaza coordonatele orizontala si verticala a cursorului,precum si starea butoanelor.

Functia 0004h permite stabilirea pozitiei curente a cursorului mouse-ului.Pentru apel se incarca in registrul ax valoarea 0004h,iar in registrii bx si cx se incarca noua coordonata orizontala,respectiv verticala.Functia nu returneaza nimic.

6.5.Functiile 0005h si 0006h

Functia 0005h permite citirea starii de apasare a butonului.Pentru apel,se incarca in registrul ax valoarea 0005h,iar in registrul bx se incarca numarul butonului dorit.Functia returneaza starea butoanelor,numarul de apasari de la ultimul apel,coordonata orizontala a cursorului la ultima apasare,precum si pe cea verticala,in aceleasi conditii.

Functia 0006h permite citirea starii de liber a butonului.Apelul se face identic cu apelul functiei 0005h,cu deosebirea ca in ax se incarca valoarea 0006h.Functia returneaza starea butoanelor,numarul de eliberari de la ultimul apel precum si coordonatele orizontala si verticala ale cursorului la ultima eliberare.

“

6.6.Functiile 0007h si 0008h

Functia 0007h permite stabilirea intervalului pentru coordonata orizontala a cursorului mouse-ului.Pentru apel se incarca in ax valoarea 0007h,in cx valoarea minima a coordonatei iar in dx valoarea maxima.Functia nu returneaza nimic.

Functia 0008h permite stabilirea intervalului pentru coordonata verticala a cursorului mouse-ului.Pentru apel se incarca in ax valoarea 0008h,in bx valoarea minima a coordonatei,iar in cx valoarea maxima.Functia nu returneaza nimic.

6.7.Functiile 0009h si 000Ah

Functia 0009h permite definirea cursorului in modul grafic.Pentru apel se incarca in ax valoarea 0009h,in registrii bx si cx coordonatele orizontala respectiv verticala ale punctului de referinta,iar in es:dx un pointer catre cele doua masti.Functia nu returneaza nimic.

Functia 000Ah permite definirea cursorului in modul text.Pentru apel se incarca in ax valoarea 000Ah,iar in bx o valoare ce specifica tipul cursorului.Daca s-a specificat cursorul software,in cx si dx se incarca masca ecranului,respectiv a cursorului,iar daca s-a specificat cursorul hardware,in cx si dx se incarca liniile de inceput,respectiv de sfarsit ale cursorului.Functia nu returneaza nimic.

6.8.Functia 000Bh

Aceasta functie permite citirea deplasarii cursorului mouse-ului.Pentru apel se incarca in ax valoarea 000Bh,iar functia returneaza numarul ratelor de deplasare pe orizontala si pe verticala.

6.9.Functia 000Fh

Aceasta functie permite definirea senzitivitatii mouse-ului.Pentru apel se incarca in ax valoarea 000Fh,iar in cx si dx valorile senzitivitatilor pe orizontala,respectiv pe verticala.Functia nu returneaza nimic.

6.10.Functia 0010h

Aceasta functie permite invalidarea cursorului intr-o zona ecran.Pentru apel se incarca in ax valoarea 0010h,in cx si dx valorile coordonatelor coltului stanga sus,iar in si si di valorile coordonatelor coltului dreapta jos.Functia nu returneaza nimic.

6.11.Functia 0012h

Aceasta functie permite definirea dimensiunilor blocului cursorului grafic.Pentru apel se incarca in ax valoarea 0012h,in bh o valoare care specifica latimea cursorului iar in ch o valoare ce specifica inaltimea cursorului.In bl se incarca o valoare care specifica coordonata orizontala a punctului de referinta,in cl o valoare care specifica coordonata verticala a punctului de referinta,iar in es:dx un pointer catre cele doua masti.Functia returneaza valoarea FFFFh in registrul ax,in caz de succes.

7.ECRANUL VIRTUAL

|  |
| --- |
| Mod Adaptor Ecran Celula Biti/pixel/plan  video virtual caracter |
| 00 C,E,P 640x200 16x8 -  01 C,E,P 640x200 16x8 -  02 C,E,P 640x200 8x8 -  03 C,E,P 640x200 8x8 -  04 C,E,P 640x200 2x1 2  05 C,E,P 640x200 2x1 2  06 C,E,P 640x200 1x1 1  07 M,E,P 640x200 8x8 -  0D E 640x200 16x8 2  0E E 640x200 1x1 1  0F E 640x350 1x1 1  10 E 640x350 1x1 1  30 P 720x350 1x1 1  - H 720x350 1x1 1 |

(Tabel de reprezentare a modurilor video )

Software-ul de mouse vede ecranul fizic al calculatorului ca pe o matrice de puncte, numita ecran virtual .Dimensiunile acestuia depind de modul video ( a se vedea tabelul de mai sus ). Driver- ul trebuie sa cunoasca in orice moment modul in care opereaza adaptorul video . El realizeaza acest lucru interceptind intreruperea 10H , prin care programele apeleaza functii video din BIOS. La fiecare apel al functiei care stabileste modul video (intreruperea 10H cu registrele AH = 0 si AL = mod ) , driver-ul de mouse determina care sunt dimensiunile ecranului virtual care trebuie utilizat .Functia 0 este singura functie din intreruperea 10H care este interceptata de catre software-ul de mouse . Daca programul utilizeaza alte metode pentru schimbarea modului ecran (de pilda ,accesul direct la registrele hardware ale adaptorului video ), este posibil ca mouse-ul sa fie “dezorientat” , iar cursorul sa fie desenat incorect sau chiar deloc . De aceea , nu este indicata folosirea metodelor nestandard pentru rezolvarea unor probleme (schimbarea sau interogarea modului video , modificarea paletei de culori pe adaptoarele EGA sau VGA ) pentru care serviciile puse la dispozitie de catre BIOS sunt suficient de puternice chiar pentru cele mai pretentioase programe .

Indiferent de modul video , driver-ul utilizeaza perechi de coordonate virtuale (X,Y) pentru localizarea oricarui obiect pe ecran. Multe functii ale driver-ului accepta coordonate virtuale ca parametri sau intorc valori care reprezinta astfel de coordonate . La apelul unor astfel de functii , trebuie sa ne asiguram ca parametrii pe care ii furnizam reprezinta coordonate valide pentru modul video curent . Valorile intoarse de functiile de mouse respecta intotdeauna aceasta conditie .

In modurile video grafice 06H , 0EH , 0FH , 10H , precum si pe adaptoarele Hercules , exista o corespondenta 1:1 intre punctele din ecranul virtual si pixeli din ecranul fizic. De exemplu in modurile grafice 4 si 5 ecranul fizic foloseste 2 biti pe pixel pentru un singur plan de memorie ecran , ceea ce face ca numarul de pixeli afisabili pe orizontala sa fie redus la jumatate . Pentru a compensa aceasta , software-ul de mouse utilizeaza numai coordonate virtuale orizontale pe care sunt numere pare .Astfel spus , numai fiecare al doilea punct din ecranul virtual corespunde efectiv unui pixel afisabil .

Modurile video 2,3 si 7 sunt moduri text , in care pe ecran pot fi afisate numai caractere . Fiecare caracter este un grup de 8x8 pixeli .Deoarece software-ul de mouse nu poate accesa pixeli individuali dintr-un caracter ,el utilizeaza coordonatele pixelului din coltul stinga -sus al cutiei caracterului .Intrucit fiecare caracter reprezinta o celula de 8x8 pixeli , atit coordonatele orizontale cit si cele verticale sunt multiplii de 8 .

In modurile text 0 si 1 , la fel ca in modurile text 2,3 si 7 , pe ecran nu se pot afisa decit caractere . Fiecare caracter reprezinta o celula de 16x8 pixeli .Software-ul de mouse utilizeaza coordonatele unui singur pixel din celula caracterului .Totusi deoarece ecranul are , pe orizontala , numai jumatate din numarul de pixeli din modurile 2,3 sau 7, software-ul foloseste numai coordonate orizontale multiplii de 16 .

8.ALTERNATIVE LA TRADITIE

Dezvoltarea exploziva a calculatoarelor,atat in domeniul software cat si in cel hardware,a dus la o expansiune a dispozitivelor de intrare.Oferta de mouse-uri a crescut foarte mult,nelimitandu-se la cazul traditional cu doua sau trei butoane.Designuri fanteziste,culori vii,modele ciudate-iata numai cateva dintre aspectele pe care le prezinta astazi oferta de mouse.

Referitor la forma oarecum standardizata a mouse-ului,firma MotorMouse a spart tabu-urile oferind utilizatorului un mouse in forma automobilului Lamborghini.Firma CompuPlus ofera la randul ei un set “Mouse Cover” care acopera mouse-ul cu o blana,ceea ce ii confera acestuia o asemanare cu confratele sau animal.

Pe piata au aparut si mouse-urile programabile.Se pare ca firmele care le produc au ca motto “cu cat mai multe butoane,cu atat mai bine”.Suntem pusi in fata unor dispozitive care nu mai prezinta sapectul stiut,cu doua sau trei butoane.Firma Amotech ofera trei modele diferite,ce pot contine pana la 40 de butoane,pe care pot fi definite diverse functii macro.Functia nu va mai fi apelata printr-o combinatie de taste,ci va putea fi activata prin apasarea unui buton de pe mouse.Exista niste fisiere KDT (Key Definition Table) care fac asocierea intre butonul apasat si o anumita functie a unui meniu,fara a se mai folosi tastatura.

Mouse-urile fara cablu pot lucra cu comanda radio sau prin infrarosu.Modelul Logitech Mouseman Cordless este unul cu comanda radio avand ca sursa de curent o baterie cu litiu,iar modelul Z-Mix Cordless Supermouse lucreaza cu infrarosu,fiind alimentat de un acumulator integrat.Avantajul mouse-ului cu comanda radio este ca receptia semnalelor provenite de la mouse nu este influentata de directie.Pentru a se reduce riscul interferentelor,se lucreaza cu patru frecvente diferite.Raza lui de actiune este de 2 pana la 3 metri.La mouse-ul cu infrarosu,emitatorul si receptorul trebuie sa se afle in contact vizual intr-o raza de 1 pana la 1.5 metri.Daca intre acestia doi se intercaleaza un obiect opac,se poate deranja receptia,cel mai uzual efect fiind aparitia unui fals “dublu-click”,iar rezultatele sunt lesne de imaginat.

Firam Gyration produce un mouse pe nume GyroPoint Desk.Acesta are butonul de click pe partea laterala stanga,si nu deasupra.Conectat la calculator printr-un port serial functioneaza,in general,ca un mouse obisnuit.Ceea ce este mai interesant este faptul ca poate fi aeropurtat.El contine un giroscop care sesizeaza miscarea si deplaseaza cursorul pe ecran.Produsul este compatibil cu Windows 3.1 si Windows 95.

9.CONCLUZII

Mouse-ul este unul dintre cele mai importante si mai usor de manevrat dispozitive de intrare.Si-a cucerit locul in top prin usurinta folosirii lui,precum si prin intuitivitatea functionarii lui.Oferta de mouse la ora actuala este enorma.Unele sunt mai usor de folosit,altele sunt ergonomice iar altele chinuie mana utilizatorului.Un accesoriu important al mouse-ului este padul de mouse.Acesta poate influenta la randul lui functionarea mouse-ului.Bila mouse-ului are aderenta mai mare pe o suprafata textila decat pe una plasticata,dar praful se acumuleaza mai bine pe suprafetele textile.Exista si probleme de dimensiuni:putem avea surpriza ca la capatul padului,cursorul sa nu se afle la capatul ecranului.

In oferta de mouse,piata este dominata de firma Logitech,urmata in ordine de firmele Microsoft,Andere si Genius.Multe tipuri de mouse apartin clasei inferioare de preturi,intre 15 si 20 DM.Un mouse ergonomic poate ajunge pana la 60 DM

Bibliografie

1.Radu Rădescu, *Echipamente periferice*, Editura Electra, Bucureşti, 2006.

2. Z. Baruch, Sisteme de intrare/iesire ale calculatoarelor, Editura Albastra, Cluj-

Napoca, 2003.

3. J.M. Feldman, Computer Architecture, A Designer’s Text Based on a Generic

RISC, McGraw-Hill, 1994

4. Scott Mueller, PC – depanare si modernizare, Editura Teora, Bucuresti, 1997.

5. Scott Mueller, Upgrading and Repairing PCs, 14th Edition, Que, 2003.

6. Winn L. Rosch, Totul despre hardware, Editura Teora, Bucuresti, 1999.

7. Adrian Sandita, Florin Constantinescu, Tiberiu Dumitrescu, Arhitecturi PC.

Notiuni hardware, Editura Aius, Cluj-Napoca, 1996

8. Andrew Tanenbaum, Organizarea structurata a calculatoarelor, editia a IV-a,

Computer Press Agora, Bucuresti, 1999.

Sandu Cristina

**GESTIUNEA DE CĂTRE UTILIZATORI**

1. **Tipuri de interfeţe de intrare/ieşire(I/O)**

Din punctul de vedere al direcției de transfer al informației, dispozitivele periferice se clasifică în:

* Dispozitive periferice de intrare (input device): tastatura, mouse, scanner, joystick, touch board, touch screen, touch pad, microfon, webcam, creion grafic, etc.
* Dispozitive periferice de ieșire (output device): monitor, imprimanta, boxe, căsti, etc.
* Dispozitive periferice de intrare/iesire (I/O device): hard disk, memorie flash, streamer, CD-RW, DVD-RW, MO-disk, RAM/static-drive, placa de retea, router, switch (comutator).

Dispozitivele care sunt folosite pentru a introduce datele și programele din computer sunt cunoscute sub numele de "Dispozitive de intrare". Un dispozitiv de intrare are funcţia de a citi datele și de a le converti într-o formă pe care un calculator poate să le utilizeze.

Dispozitivele de ieșire transformă produsul final al prelucrării mașinii într-o formă utilizabilă de către om. Acesta oferă comunicarea computerului cu utilizatorul.

**Din punct de vedere funcțional, dispozitivele periferice se împart în:**

* dispozitive periferice de schimb: imprimanta, plotter-ul, rețelele de comunicare cu alte periferice;
* dispozitive periferice purtătoare de informatii permanent pe medii magnetice: benzile magnetice, discurile (hard-discurile, dischetele);

Din punctul de vedere al tipului de transfer al informației, avem:

* dispozitive periferice bloc;
* dispozitive periferice caracter;

Diferența dintre acestea se poate observa datorită literei care apare pe prima coloană în ieșirea comenzii ls-1, această coloană indică tipul fișierului. Fișierele care au tipul C indică dispozitivele de tip caracter, iar cele care au tipul B indică dispozitivele de tip bloc.

Unele dintre dispozitive sunt generate de kernel si au componente aparte. Acestea se numesc pseudo-dispozitive.

Dispozitivele caracter corespund dispozitivelor care transmit câte un caracter odată. Acestea trimit sau primesc un șir de caractere(stream) fără nicio organizare a informației pe adrese, de aceea nu se poate accesa o anumită porțiune.

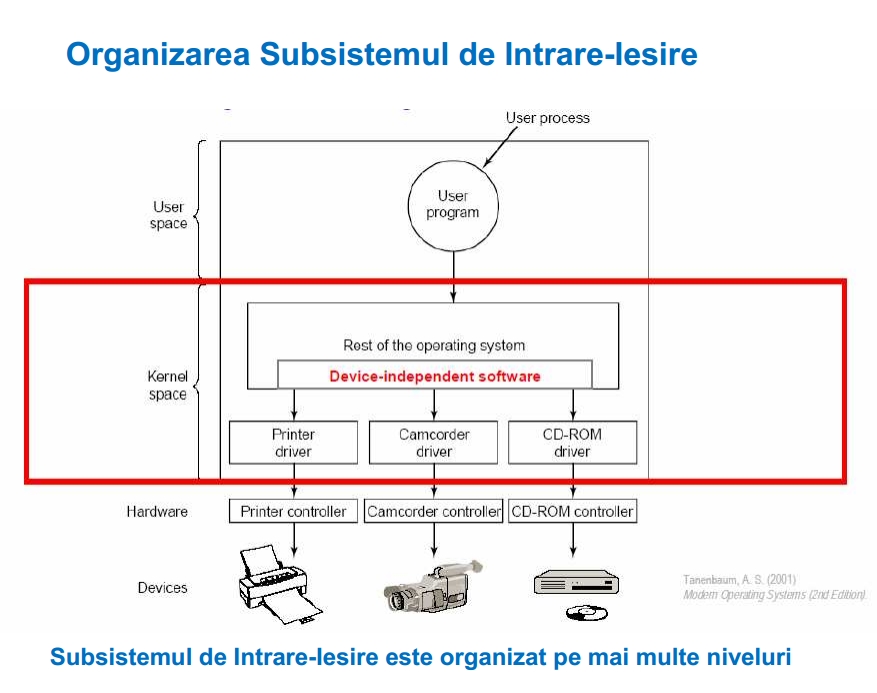
Dispozitivele bloc corespund dispozitivelor care lucrează cu datele la nivel de bloc (citesc,scriu,sterg blocuri de date). Acestea memorează informația în blocuri de dimensiuni fixe (512 la 32,768byte) fiecare cu adresa proprie. Caracteristica fundamentală a acestora este ca informația poate fi citită doar din blocurile care interesează, desigur pentru un cost de timp (ex: hard-disk).

Dispozitivele care funcționeaza astfel sunt cele de stocare:

* Fizice - unităti CD-ROM/DVD-ROM, hard-diskuri
* Virtuale – dispozitive loop
* Asociate zonelor de memorie

Dispozitivele de tip bloc sunt mult mai performante decât cele de tip caracter, deoarece viteza celor de tip bloc este mult mai mare decât celelalte. Importanța acesteia este foarte mare, pentru ca Linux tratează diferit cele două tipuri de dispozitive. Pe când ,Windows, prin intermediul I/O managerul oferă o interfața unificata pentru tratarea dispozitivelor.

Lucrul cu dispozitivele de tip caracter e mult mai ușor față de celelalte. Acestea au o singură pozitie curentă, în timp ce cele de tip bloc trebuie să se poată mișca la orice poziție din dispozitiv pentru asigurarea accesului aleator la date. Pentru a simplifica lucrurile cu dispozitivele de tip bloc, nucleul Linux pune la dispozitie un întreg subsistem denumit subsistemul block I/O (block layer).



Link: http://www.cs.ucv.ro/staff/cpatrascu/SIE/Structura%20sistemului%20IE.pdf

Pseudo-dispozitivele (ceasurile) sunt o serie de dispozitive care nu corespund unei componente hardware. Acestea pot fi echivalente unor dispozitive virtuale care implementează comportamentul dispozitivelor normale de tip caracter.

1. **Modalităţi de transfer de intrare/ieşire**

Din punctul de vedere al circuitelor care controlează transferul şi de asemenea al dispozitivului care iniţiază un transfer de I/O, modalităţile de transfer se pot clasifica în:

1. transfer prin program - transfer iniţiat şi controlat în totalitate de programul rulat de UCP;

2. transfer prin întreruperi - transferul este controlat de UCP ca răspuns la o cerere de întrerupere externă, care iniţiază transferul;

3. transfer prin acces direct la memorie (DMA - "Direct Memory Acces") - transferul este controlat de un circuit controler DMA (care preia controlul magistralelor sistemului), iar iniţierea transferului este făcută fie de o cerere de transfer de la un periferic, fie la iniţiativa programului rulat de UCP;

* 1. **Transferul de I/O prin program**

La acest tip de transfer, toate transferurile se iniţiază şi se controlează prin program. Transferul poate fi direct, caz în care procesorul citeşte un port de intrare (PI) sau scrie un port de ieşire (PO), fără nici o verificare prealabilă a stării perifericului corespunzător portului. Acest mod de transfer se foloseşte pentru sistemele simple de control numeric, dedicate unor aplicaţii fixe, în faza de proiectare şi testare a transferului (program plus hardware), sau în cazul rulării unor rutine de iniţializare a circuitelor programabile de interfaţă.

Al doilea mod de transfer prin program este transferul prin interogare. Intrarea în conversaţie cu un periferic se face sub controlul programului, de obicei într-o buclă de interogare (polling - scrutare a stării circuitelor periferice implicate în transfer). UCP interoghează dispozitivele periferice, după o anumită strategie stabilită prin program, dacă doresc sau nu schimb de date sau mesaje, sau dacă sunt active (gata pentru a primi informaţii). Pentru a determina dacă o operaţie de I/O este cerută, sau dacă poate avea loc, se pot folosi bistabile de condiţie (fanioane) locale, setate conform condiţiilor portului. Aceste fanioane se implementează fizic fie ca bistabile singulare, fie sunt incluse în registre de stare ale porturilor. Datorită simplităţii ei, tehnica de comunicare prin interogare programată este recomandată în sistemele în care timpul pierdut în rutina de interogare nu este critic pentru aplicaţia respectivă.

Atunci când programul are de controlat mai multe periferice, pe baza unei liste a porturilor asociate perifericelor, procesorul testează pe rând starea perifericelor şi le serveşte din punctul de vedere al transferului, dacă starea citită permite acest lucru. În acest fel perifericele sunt deservite în mod secvenţial şi implicit apar întârzieri la servirea acestora, mai ales dacă numărul de periferice controlate este mare. Dacă unele dintre periferice se consideră mai importante decât altele, adresa acestora se poate introduce de mai multe ori în lista de testare din bucla de interogare.

Dezavantajul principal al modului de transfer programat este constituit de timpul pierdut de UCP pentru testarea stării perifericelor, chiar dacă acestea nu cer servicii la momentele de timp respective.

Acest mod de transfer prezintă însă avantaje din punctul de vedere al costurilor (minim de echipament fizic suplimentar pentru transfer). Alt avantaj este că se cunoaşte exact momentul de timp când se testează sau când se face transfer de date cu un periferic; de aceea se spune că acest tip de transfer este sincron cu programul.

* 1. **Transferul I/O prin întreruperi**

Evenimentele externe procesorului, deci independente de programul rulat, pot produce cereri de întrerupere. Dacă acestea sunt acceptate, se produce întreruperea (suspendarea temporară a) programului rulat şi saltul la o rutină specifică de tratare a cererii de întrerupere. După execuţia rutinei de tratare se revine la execuţia programului întrerupt.

Subsistemul de întreruperi al calculatorului nu este destinat special doar pentru operaţii de transfer de I/O, dar aici ne vom referi doar la acest aspect. Din punctul de vedere al transferurilor de I/O, avantajele sistemului de întreruperi sunt următoarele:

•permite sincronizarea procesorului cu evenimente externe;

•eliberează procesorul de sarcina testării periodice a perifericelor, sarcină consumatoare de timp. Ca urmare transferul prin întreruperi prezintă o viteză de răspuns mai mare decât transferul prin program;

• posibilitatea de tratare ierarhizată a cererilor de întrerupere simultane sau succesive;

Cererile de întrerupere pot fi recunoscute doar la sfârşitul ciclului instrucţiune curent. În general, pentru efectuarea transferurilor de I/O se folosesc cererile de întrerupere mascabile. Răspunsul UCP la acceptarea unei cereri de întrerupere mascabilă, constă în următoarele operaţii succesive:

1. Salvarea stării programului întrerupt. În această fază se salvează automat în stivă cel puţin adresa de revenire la programul întrerupt. Alte informaţii privind starea programului se pot salva prin instrucţiuni introduse la începutul rutinei de tratare, iar înainte de revenire la programul întrerupt trebuie introduse instrucţiuni pentru restaurarea acestor informaţii.

2. Confirmarea acceptării şi identificarea întreruptorului. Confirmarea acceptării întreruperii se face prin transmiterea de UCP a unui semnal de acceptare, în urma căruia perifericul trimite, de obicei, un vector de întrerupere pentru identificare. Există şi sisteme de identificare care nu folosesc vector de întrerupere.

3. Calcularea adresei de început a rutinei de tratare a întreruperii. Pe baza informaţiei de identificare se calculează unde se va face saltul pentru ca cererea de întrerupere, pentru transferul de date, să fie servită.

4. La terminarea rutinei de tratare, se execută o instrucţiune specială, care comanda refacerea conţinutului contorului de program şi deci revenirea la programul întrerupt.

Generarea vectorului de întrerupere se face fie direct de către dispozitivul întreruptor (de fapt de către circuitul de interfaţă cu un anumit periferic), fie centralizat, pentru toate întreruperile mascabile, de către un dispozitiv special numit controller de întreruperi.

**Arbitrarea întreruperilor multiple**.

În cazul întreruperilor multiple, simultane sau succesive, subsistemul de întreruperi trebuie să realizeze un arbitraj al acestora pentru a stabili care dintre cererile de întrerupere multiple va fi servită la un moment dat. Sistemul de arbitrare alocă priorităţi cererilor de întrerupere, servind întotdeauna cererile cu prioritatea cea mai mare. Se permite, de asemenea ca o rutină de tratare să poată fi la rândul ei întreruptă de o cerere cu prioritate mai mare. De obicei, pentru un procesor de uz general, priorităţile intrinseci sunt atribuite în următoarea ordine: întreruperile interne (excepţiile) cu prioritatea cea mai mare, apoi urmează întreruperile externe (hardware) nemascabile şi în final cele mascabile. Arbitrarea întreruperilor multiple se poate realiza prin mai multe metode:

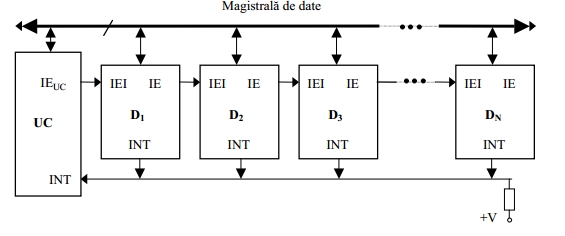
Arbitrare :

* controlată de UCP- prin software, prin hardware;
* controlată de circuit controler de întreruperi;
* controlată prin hardware, în lanţ de priorităţi;

În cazul arbitrării controlate de UCP prin hardware, UCP conţine intern circuitele necesare realizării arbitrării. Circuitul de arbitrare primeşte mai multe intrări de cereri de întrerupere, acestea, sau combinaţii ale acestora, corespunzând la un anumit nivel prefixat de prioritate. Adresele rutinelor de tratare pot fi fixe (cum este de exemplu cazul microprocesorului I8085) sau variabile (de exemplu la microprocesorul MC68000).

Circuitele au ca element principal un registru-port de intrare, în care se alocă câte un bit pentru fiecare dispozitiv întreruptor. Dacă acest bit este setat, dispozitivul a cerut cerere de întrerupere. Simultan cu setarea bitului din registru, semnalul de cerere de întrerupere se transmite şi către UCP. Prin citirea registrului port de intrare şi compararea cu situaţia anterioară (memorată într-un registru intern), UCP detectează dacă a apărut o cerere de întrerupere cu prioritate mai mare decât cea a programului curent. În cazul unei întreruperi cu prioritate mai mică se continuă programul curent (după eventuala înregistrare a cererilor în vederea servirii ulterioare). În cazul priorităţii mai mari, UCP începe execuţia subrutinei de servire asociată noii cereri, întrerupând astfel programul curent şi actualizând registrul de serviciu. După terminarea execuţiei subrutinei de servire se reia ultimul program întrerupt. De asemenea, după comutarea contextului, UCP trebuie să revalideze întreruperile pentru a permite unor eventuale cereri de întrerupere prioritare să fie luate în considerare. Cererile de întrerupere ale dispozitivelor individuale de I/O, pot fi activate sau dezactivate prin program, prin intermediul unor bistabili de mască, asamblaţi de obicei într-un registru de mascare. Astfel că la momente diferite de timp, prin aplicarea unor măşti diferite, programul poate modifica ordinea de (prioritate de) servire a cererilor de întrerupere.

Metoda de arbitrare descentralizată prin lanţ de priorităţi a întreruperilor presupune existenţa în fiecare dispozitiv întreruptor a unei logici capabile să asigure protocolul electric de tratare a întreruperilor. Dispozitivele întreruptoare se conectează într-un "lanţ de priorităţi" (daisy chain), ca în figura:



Exemplu de arbitrare a cererilor multiple de întrerupere prin lanţ de priorităţi

Link: <http://vega.unitbv.ro/~romanca/AOC/Cap7-AOC-IntrareIesire.pdf>

UCP generează semnalul de validare a întreruperilor externe (IEUCP) care se propagă prin lanţ, trecând prin celulele succesive ale lanţului. Pentru fiecare dispozitiv, starea intrării de validare IEI (Interrupt Enable Input) este copiată la ieşirea IEO (Interrupt Enable Output) a fiecărui dispozitiv, dacă dispozitivul respectiv nu cere întrerupere către UCP. Dacă un dispozitiv a cerut întrerupere, care a fost recunoscută şi se face tratarea acesteia, pe tot timpul servirii între intrarea IEI şi ieşirea IEO ale dispozitivului servit nu mai există egalitate. Pe toată această perioadă dispozitivul aduce IEO la zero pentru a nu permite dispozitivelor cu prioritate mai mică să-i întrerupă rutina de tratare.

În momentul în care un dispozitiv Di cere întrerupere, pot exista două situaţii:

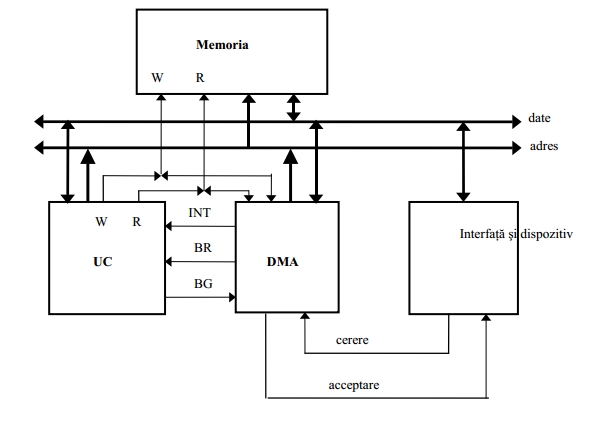
⇒ dacă IEIi=0, atunci un dispozitiv mai prioritar decât Di a cerut întrerupere şi servirea sa de către rutina corespunzătoare nu s-a încheiat. Ca urmare dispozitivul Di va aştepta până când IEIi trece în stare activă "1".

⇒ dacă IEIi=1, atunci Di poate genera cererea de întrerupere către INT (aducând această linie la zero logic); aceasta pentru că Di este dispozitivul cel mai prioritar care cere întrerupere la momentul respectiv de timp. Ca urmare, după lansarea cererii de întrerupere, IEOi devine zero, invalidând cererile de întrerupere de la dispozitivele cu prioritate mai mică din lanţ. Din cele de mai sus rezultă că dacă a apărut o cerere de întrerupere către UCP, în lanţ, un singur dispozitiv are IEI = 1 şi IEO = 0, şi anume cea servită. Pentru toate celelalte dispozitive ieşirea IEO are valoare logică identică cu intrarea IEI. Dacă UCP acceptă cererea de întrerupere, dispozitivul întreruptor depune pe magistrala de date a sistemului vectorul de întrerupere, pentru a fi identificat şi servit corespunzător.

* 1. **Transferul prin acces direct la memorie(DMA)**

Transferul, prin intermediul UCP, a blocurilor mari de date între memoria principală şi periferice, se face relativ lent, pentru că de fiecare dată informaţia trece şi prin registre ale UCP. În plus pentru o cantitate mare de date transferate între memorie şi periferic UCP va consuma foarte mult timp. Transferul prin acces direct la memorie (DMA - Direct Memory Acces) este executat direct între memoria principală şi dispozitivul periferic, fără ca datele să mai treacă prin UCP, sub comanda unui circuit controler DMA, care controlează temporar magistralele sistemului. În figura se presupune că iniţierea transferului se face de către periferic (prin intermediul interfeţei) care efectuează o cerere de transfer prin acces DMA spre circuitul controler de DMA (DMAC). Acesta solicită de la UCP controlul magistralelor prin semnalul de cerere de control a magistralelor, BR, (Bus Request). Cu o mică întârziere (la sfârşitul ciclului maşină curent) UCP cedează controlul magistralelor, îşi trece ieşirile către acestea în HiZ (stare de înaltă impedanţă) şi informează despre aceasta prin semnalul acordare a controlului magistralelor, BG, (Bus Grant).

Circuitul controler DMAC furnizează adresele pe magistrala de adrese, preia controlul semnalelor de scriere (WR) şi citire (RD) şi trimite către periferic semnalul de acceptare a transferului prin DMA. Când dispozitivul I/O primeşte acest semnal de acceptare, el pune un cuvânt pe magistrala de date (pentru scriere în memoria principală) sau citeşte un cuvânt de pe magistrala de date (pentru citire din memoria principală).



Exemplu de transfer DMA (DMAC = Controller DMA)Link: <http://vega.unitbv.ro/~romanca/AOC/Cap7-AOC-IntrareIesire.pdf>

Transferul de date se va face direct între periferic şi memoria principală, fără ca UCP să mai fie intermediar al acestui transfer (aşa cum se întâmpla la transferul prin program şi prin întreruperi).

Un ciclu DMA reprezintă transferul unui cuvânt din sau în memorie. După terminarea unui ciclu, controlerul DMA poate continua cu alte cicluri DMA (dacă a fost programat să facă acest lucru şi dacă cererea DMA de la periferic se păstrează activă), sau poate returna controlul magistralelor către UCP prin inactivarea semnalului BR. Pentru transfer pe cuvânt, secvenţa de semnale de cerere DMA (cerere → BR → BG → acceptare) conduce la desfăşurarea unui singur ciclu DMA.

Pentru un alt ciclu DMA, secvenţa de semnale trebuie repetată şi abia apoi se transferă un nou cuvânt. Cealaltă variantă de transfer, în mod rafală-bloc, presupune transferul mai multor cuvinte pentru fiecare cerere de transfer DMA. Este un mod de mare viteză, care însă dacă controllerul şi UCP sunt legate la aceleaşi magistrale poate menţine procesorul în inactivitate pentru o perioadă de timp. În cazul transferului pe cuvânt, întreruperea poate fi insesizabilă, ca timp, pentru procesor, şi de aceea uneori acest mod de transfer este numit "cu furt de ciclu".

De obicei circuitele controler DMA pot gestiona transferuri cu mai multe periferice simultan, pentru fiecare periferic existând (intern controllerului) un canal DMA.

Structura controlerului DMA trebuie să cuprindă:

* Logică de comandă şi sincronizare. Modul de lucru al controllerului DMA este programabil. De aceea el este iniţializat printr-o rutină de iniţializare rulată de UCP, fixându-se astfel: modul de transfer (cuvânt sau bloc-rafală), sensul transferului (la memoria principală sau de la memoria principală), nivelul activ al semnalelor de interfaţă cu echipamentul periferic, priorităţile acordate canalelor, modul de tratare al sfârşitului de transfer etc. Circuitul controller DMA are de asemenea o interfaţă cu UCP, care permite iniţializarea registrelor controlerului, comanda ulterioară de către UCP a controlerului, precum şi realizarea protocolului BR/BG;
* Circuite tampon pentru conectarea la magistralele calculatorului;
* Logică de arbitrare a priorităţilor canalelor;
* Logică specifică fiecărui canal DMA, care cuprinde cel puţin următoarele componente:

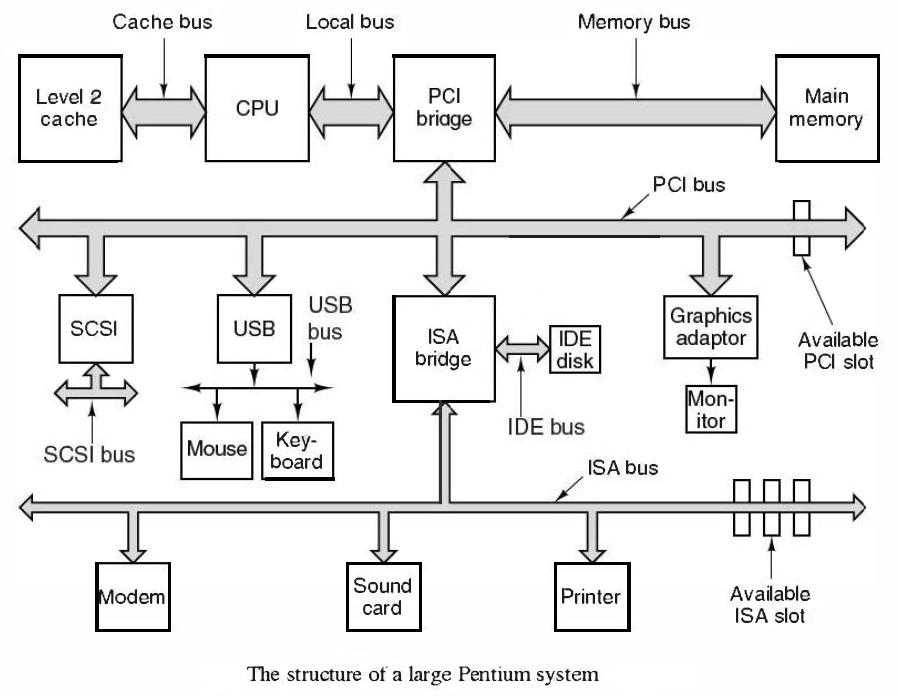
⇒ un registru numărător de adrese care generează adresa curentă (din memoria principală) de transfer;

⇒ un numărător de cuvinte de transferat, care se decrementează la fiecare ciclu DMA efectuat (valoarea iniţială a acestui contor este egală cu numărul de cuvinte ce se doreşte a fi transferat. Atunci când conţinutul a ajuns la zero, ciclul de transfer s-a sfârşit);

⇒ un registru de stare a canalului, care poate fi citit de UCP şi care indică: sensul transferului prin canal, canal activat / dezactivat, prioritatea alocată etc.

Transferul prin DMA prezintă avantaje din punctul de vedere al vitezelor mari de transfer, pentru blocuri mari de date. Este un transfer folosit în aplicaţii de genul: transfer cu discurile magnetice, transfer cu plăci periferice ce conţin convertoare AD sau DA rapide etc.

1. **Magistrala de intrare/iesire**

****

Link: http://www.osinfoblog.com/post/21/buses/

Magistrala I/O permite procesorului sa comunice cu dispozitivele periferice. Magistrala I/O permite adaugarea de dispozitive calculatorului pentru a-i extinde posibilitatile. Numarul sloturilor de extensie poate sa varieze de la un calculator la altul.

***Tipuri de magistrale I/O***

În scopul obtinerii unor performante cât mai bune, respectiv cresterea vitezei de intrare/iesire a fost necesara si îmbunatatirea magistralelor I/O, impunându-se ca acestea sa fie cât mai rapide. În functie de arhitectura lor, pot fi identificate urmatoarele tipuri de magistrale de I/O:

-         ISA (Industry Strandard Arhitecture)

-         MCA (Micro Channel Arhitecture)

-         EISA (Extended ISA)

-         VL-Bus (VESA Local Bus)

-         PCI (Peripheral Component Interconnect)

-         AGP

Aceste magistrale se deosebesc în principal prin volumul datelor transferate simultan si prin viteza cu care se face acest transfer. Arhitectura magistralei este realizata cu un set de cipuri care este conectat la magistrala procesorului.

**Magistrala ISA** este arhitectura de magistrala utilizata la primul IBM PC în 1981. Initial a fost o magistrala de 8 biti, iar apoi a fost extinsa la 16 si 32 de biti. ISA sta la baza calculatorului personal modern si este principala arhitectura folosita astazi la marea majoritate a sistemelor PC. Ea a permis ca majoritatea componentelor unui sistem, cum ar fi porturile seriale, porturile paralele, adaptoarele video si de sunet sa fie interschimbabile între sistemele compatibile IBM.

În cazul*magistralei ISA de 8 biti*, o placa adaptoare are 62 de contacte pe marginea sa inferioara, ea se introduce într-un conector cu 62 de contacte montat pe placa de baza. Din punct de vedere electronic, acest conector furnizeaza 8 linii de date si 20 de linii de adrese si permite lucrul cu 1 M de memorie.

*Magistrala ISA pe 16 biti*a aparut pe piata odata cu lansarea procesorului 286. Acest procesor avea o magistrala de date care îi permitea comunicatia pe 16 biti cu placa de baza si cu memoria. Din acest motiv a fost necesara fie crearea unei noi magistrale I/O, cu sloturi corespunzatoare de extensie, fie proiectarea unui sistem care sa admita atât placile pe 8 biti cât si cele pe 16 biti. IBM a optat pentru cea de-a doua solutie si a introdus un sistem PC AT cu un set de sloturi de extensie care utilizau conectori suplimentari pentru lucrul pe 16 biti. Conectorul suplimentar al slotului de extensie contine 36 de pini pentru semnale necesare realizarii unei cai mai largi de date. În plus, doi dintre pinii conectorului de 8 biti au primit o alta destinatie.

*Magistrala ISA pe 32 biti*a fost disponibila dupa o perioada de la aparitia procesoarelor pe 32 biti. Portiunea suplimentara a magistralei era folosita în general pentru extensii de memorie sau placi video particulare.

**Magistrala MCA**. Aparitia procesoarelor pe 32 de biti a facut ca magistrala ISA sa nu mai corespunda puterii noii generatii de procesoare. În loc sa extinda magistrala ISA, IBM a decis sa construiasca o noua magistrala MCA (Micro Channel Architecture). MCA este complet diferita de magistrala ISA si îi este superioara din toate punctele de vedere. Un sistem MCA este foarte usor de folosit deoarece este lipsit de jumpere si comutatoare, atât pe placa de baza, cât si pe placile adaptoare. Acest tip de magistrala  nu a fost utilizat pentru mult timp datorita faptului ca placile adaptoare pentru sisteme ISA nu functioneaza în sisteme MCA.

MCA admite controlul total al magistralei (bus mastering) prin care orice dispozitiv poate cere acces la magistrala în scopul comunicarii cu un alt dispozitiv, conectat si el la magistrala. Aceasta cerere se face prin intermediul unui dispozitiv CACP (Central Arbitration Control Point). Fiecare dispozitiv are un cod de prioritate, ceea ce asigura pastrarea ordinii în sistem.

În proiectarea magistralelor MCA se folosesc patru tipuri de conectoare: pe 16 biti, pe 16 biti cu extensii video, pe 16 biti cu extensii de memorie, pe 32 de biti.

Extensiile pentru memorie marita permit utilizarea placilor de memorie îmbunatatite si transferuri de date cu aceste placi.

Extensiile video MCA sunt conectoare standard pe 16 biti însotite de câte un conector special pentru extensia video. Acest slot apare aproape în toate sistemele MCA în scopul maririi vitezei subsistemului video.

**Magistrala EISA** (Extended Industry Standard Arhitecture) permite, comparativ cu arhitectura sistemelor ISA pe 16 biti, o dezvoltare mai mare a sistemului, cu mai putine conflicte între adaptoare. Magistrala EISA adauga 90 de conexiuni suplimentare (55 de semnale noi) fara ca aceasta sa implice cresterea dimensiunilor conectorului de magistrala ISA pe 16 biti. Totusi, adaptorul EISA are doua rânduri de conectori. Primul rând este asemeni cu cel folosit de placile ISA pe 16 biti, iar al doilea, mai îngust, constituie extensia fata de conectorul de 16 biti. Sistemele EISA au un program de configurare (Setup) care gestioneaza întreruperile de la placile adaptoare si iesirile de adrese. Aceste iesiri creeaza adesea probleme atunci când sunt instalate mai multe placi adaptoare diferite în acelasi sistem.

PCI a fost inventat de Intel ca un succesor pentru magistrala ISA. . Cele mai multe dispozitive I/O de mare viteză utilizează magistrala PCI acum. Chiar și unele calculatoare non-Intel folosesc bus-ului PCI, datorită numărului mare de carduri I/O disponibile pentru el.

Magistrala PCI este conectată la magistrala locala a procesorului prin intermediul unei punţi UCP/PCI, avănd acces direct la memoria principala. În acest fel, transferurile între UCP şi memoria cache de nivel 2, respectiv între dispazitivele de I/E şi memoria principală, pot avea loc simultan. Dispozitivele de I/E de viteză ridicată, ca adaptoarele grafice şi adaptoarele de reţea, care solicită în măsură redusă procesorul, sunt conectate direct la magistrala PCI. Dispozitivele care trebuie să se conformeze altor standarde de magistrală, ca ISA sau SCSI, se interfatează cu magistrala PCI printr-o punte PCI/ISA, respectiv un adaptor SCSI.

Deoarece magistrala PCI nu este specifică procesoarelor din familia Intel, aceasta poate fi utilizată şi pentru alte procesoare. Un sistem poate fi realizat şi fără utilizarea punţii de legătură. În acest caz, toate componentele, inclusiv procesorul şi memoria principală, se interfaţează direct cu magistrala PCI. Magistrala PCI poate fi extinsă pentru a permite conectarea unui număr mai mare de plăci de extensie cu ajutorul unor punţi PCI-PCI. Atunci cănd traficul este local pe fiecare magistrală, pot fi active mai multe magistrale în acelaşi timp, ceea ce permite echilibrarea încărcării acestora. Există variante ale magistralei PCI, ca PCI-X sau PCI Expres.

Ceasul magistralei PCI este separat de ceasul procesorului. La versiunea 2.0, frecvenţa maximă a ceasului este de 33 MHz. Cu o magistrală de 32 de biţi, rata de transfer maximă este de 132 MB/s. Începănd cu versiunea 2.1, specificaţiile conţin extensii opţionale, care permit creştere ratei de transfer şi utilizarea unor platforme şi arhitecturi multiple.

Una din extensii este cea de 64 de biţi pentru date sau adrese, sau pentru ambele. Extensia de 64 de biţi pentru adrese permite extinderea spaţiului de adresare peste limita de 4 GB. Extensia de 64 biţi pentru transferurile de date permite transferul a 8 octeţi în fiecare fază de date la un dispozitiv destinaţie de 64 de biţi, rezultând o rată maximă de transfer de 264MB/s. Transferurile se vor efectua pe 32 de biţi dacă dispozitivul destinaţie selectat nu permite extensia de 64 de biţi. Extensia de 64 de biţi utilizează un conector suplimentar.

O altă extensie specificată începând cu versiunea 2.1. este cea de 66 MHz, care permite creşterea ratei de transfer pentru aplicaţiile grafice avansate şi cele video. Utilizarea acestei frecvenţe este limitată, deoarece pentru aceasta toate dispozitivele conectate la magistrală trebuie să funcţioneze la 66 MHz. Dacă unul din dispozitive de 33 MHz se conectează la magistrala de 66 MHz, magistrala va funcţiona la 33 MHz. De aceea, utilizarea magistralei PCI de 66 MHz, care este partajată de mai multe dispozitive, va creşte costul acelor dispozitive care nu necesită rate de transfer foarte ridicate, pentru ca magistrala să funcţioneze la 66 MHz.

Implementările magistralei PCI care utilizează extensia de 64 de biţi şi cea de 66 MHz permit obţinerea unei rate de transfer maxime de 528 MB/s

Toate plăcile de extensie PCI sunt configurate automat pentru tranzacţiile pe magistrală, fără a fi necesară setarea manuală a adreselor porturilor de I/E, a nivelului de întrerupere sau a canalului DMA. Prin includerea caracteristicilor OpenBoot (IEEE 1275), orice dispozitiv compatibil OpenBoot se poate utiliza pentru iniţializarea sistemelor conţinând orice procesor, fără ajutorul sistemului de operare sau a unor rutine specifice procesorului.

Magistrala PCI dispune de facilităţi avansate, cum este cea care asigură coerenţa memoriilor cache în cazul existenţei mai multor dispozitive master, sau mecanismul pentru asigurarea sincronizării prin semafoare. În acelaşi timp, magistrala permite utilizarea unor plăci de extensie sau controlere care se conformează cu standarde mai vechi. De exemplu, este posibilă utilizarea cu magistrala PCI a plăcilor de extensie ISA, prin faptul că magistrala permite decodificarea substractivă a adreselor (adresele care nu sunt decodificate de plăcile PCI fiind utilizate de o altă magistrală, cum este ISA). De asemenea, magistrala PCI permite controlerelor conforme cu standardele VGA şi IDE. Pentru aceastea, magistrala PCI permite ca aceste controlere să utilizeze adrese fixe de I/E, necesare pentru încărcarea cu succes a sistemului de operare.

Caracteristici ale magistralei PCI

- Dispozitiv iniţiator şi ţintă

La fiecare transfer participă două dispozitive: iniţiatorul şi ţinta. Iniţiatorul, sau dispozitivul master, este cel care iniţiază un transfer. Ţinta, sau dispozitivul slave, este adresat de iniţiator în scopul executării transferului. Dispozitivele iniţiator şi ţintă sunt numite agenţi PCI.

- Transferuri în mod exploziv

Un asemenea transfer constă dintr-o singură fază de adrese urmată de mai multe faze de date. Arbitrajul de magistrală trebuie executat o singură dată. În timpul fazei de adrese se transmite adresa de inceput şi tipul tranzacţiei care urmează. Dispozitivul ţintă memorează adresa de început într-un contor de adrese, şi va incrementa adresa în fiecare fază de date.

În cazul magistralei PCI, cele mai multe transferuri se execută în mod exploziv, cele mai multe dispozitive fiind proiectate astfel încât să permită asemenea transferuri. Dacă un dispozitiv ţintă poate executa numai tranzacţii singulare, la incercarea dispozitivului master de a executa o tranzacţie în mod exploziv, ţinta termină tranzacţia după prima fază de date. Astfel dispozitivul master va fi obligat să solicite din nou magistrala printr-un proces de arbitraj. În acest mod performanţele vor fi mai reduse, dar tranzacţiile singulare se pot utiliza pentru dispozitivele care nu necesită rate ridicate de transfer.

Presupunând că nici iniţiatorul nici ţinta nu inserează stări de asteptare, poate fi transferat câte un cuvânt la fiecare front crescător al semnalului de ceas. La o frecvenţă de 33 MHz, se poate obţine o rată de transfer de 132 MB/s în cazul implementării pe 32 de biţi, sau 264 MB/s pentru 64 de biţi. Cu o frecvenţă a ceasului de 66 MHz, se pot obţine rate de transfer de 264 MB/s, respectiv 528 MB/s.

- Dispozitive multifuncţionale

Un dispozitiv PCI poate fi o componentă integrată pe placa sistem sau o placă de extensie PCI. Fiecare dispozitiv poate încorpora până la opt funcţii separate, un asemenea dispozitiv fiind numit multifuncţional.

- Ceasul magistralei PCI

Frecvenţa semnalului de ceas poate fi intervalul 0-33MHz. Specificaţiile versiunii 1.0 indică faptul că toate dispozitivele trebuie să funcţioneze între 16 şi 33 MHz. Frecvenţa ceasului se poate modifica în orice moment şi ceasul poate fi oprit. Versiunile 2.1, 2.2 definesc de asemenea funcţionarea la frecvenţe de până la 66MHz.

Variante ale magistralei PCI

PCI-X 1.0 şi 2.0

PCI-X este o extensie cu performanţe mai ridicate a magistralei PCI convenţionale, compatibilă cu diversele variante ale acesteia. Această extensie asigură ratele ridicate de transfer necesare pentru conexiuni cum sunt Gigabit Ethernet, Fibre Chanel şi Ultra-3 SCSI. Actualmente, magistrala PCI-X este utilizată pentru servere şi unele staţii de lucru.

Versiunea 1.0 a magistralei PCI-X a fost standardizată de grupul PCI-SIG în anul 1999. Această versiune permite frecvenţe de funcţionare de până la 133 MHz, numărul de linii ale magistralei fiind de 32 sau 64. Pentru magistrala de 64 de biţi, rata de transfer maximă care se poate obţine la această frecvenţă este de 1064 GB/s.

Pe lîngă frecvenţa mai ridicată, versiunea 1.0 a magistralei PCI-X introduce şi unele îmbunătăţiri ale protocolului magistralei PCI convenţionale pentru o funcţionre mai eficientă, inclusiv la frecvenţe mai reduse. Principalele îmbunătăţiri sunt următoarele:

- tranzacţiile divizate permit unui dispozitiv iniţiator să efectueze o singură cerere pentru un transfer de date şi apoi să elibereze magistrala, în locul interogării continue a magistralei pentru detectarea răspunsului la această cerere.

- introducerea unui contor de octeţi permite unui iniţiator să specifice în avans numărul de octeţi solicitaţi, eliminând ineficienţa încărcărilor speculative.

- ordinea relaxată a tranzacţiilor îmbunătăţeşte performanţele în cazul sistemelor multiprocesor complexe şi aplicaţiile în timp real.

Magistrala PCI-X 1.0 asigură compatibilitatea hardware şi software cu versiunile anterioare.

Versiunea 2.0 a specificaţiilor magistralei PCI-X 2.0 introduce îmbunătăţiri suplimentare ale magistralei PCI-X păstrând compatibilitatea hardware şi software cu generaţiile anterioare ale magistralei PCI. Sunt introduse două noi frecvenţe de funcţionare ale magistralei, de 266 MHz şi 533 MHz. Variantele corespunzătoare sunt denumite PCI-X 266 (sau DDR double data rate ) şi PCI-X 533 (sau QDR quad data rate ), datele fiind transmise la o frecvenţă de 2 ori respectiv de 4 ori mai mare decât frecvenţa semnalului de ceas. Ratele maxime de transfer sunt de 2128 GB/s respectiv 4264 GB/s.Pe lîngă aceste două frecvenţe este posibilă funcţionarea şi la frecvenţe mai scăzute de 66, 100 sau 133 de MHz.

|  |
| --- |
| Pe lânga frecventele mai ridicate de functionare, magistrala PCI-X 2.0 introduce si alte carac-teristici noi, cele mai importante din acestea fiind descrise în continuare: |

|  |
| --- |
|  |

|  |
| --- |
| - utilizarea unui cod corector de erori ECC (Error Correcting Code)  îmbunatateste robustetea interfetei. Se utilizeaza 8 biti suplimentari de control, care permit corectarea erorilor de un singur bit si detectia erorilor de 2 biti. Erorile de un bit sunt corectate automat, în timp ce ero-rile de doi biti sunt marcate pentru a realiza retransmisia datelor. |

|  |
| --- |
| - introducerea unor registre de configuratie specifice versiunii 2.0 a fost necesara pentru a im-plementa functia de corectie a erorilor. |

|  |
| --- |
| - utilizarea unui protocol îmbunatatit care creste gradul de utilizare si eficienta magistralei. |

|  |
| --- |
| - o alta îmbunatatire se refera la introducerea unor semnale de strob pentru variantele PCI-X266 si PCI-X 533 ale magistralei. Aceste semnale comanda intrarile de ceas ale bufferelor de date si asigura memorarea datelor la momente precise de timp. Deoarece semnalele de strob si cele de date sunt supuse acelorasi variatii de tensiune, temperatura etc., ele deviaza în aceeasi directie si cu aceeasi amplitudine, ceea ce permite memorarea sigura a datelor si la ratele foarte ridicate ale variantelor PCI-X 266 si PCI-X 533. |

|  |
| --- |
| - utilizarea unor semnale de 1,5 V permite functionarea la frecvente mai ridicate. Pentru a mentine compatibilitatea cu tehnologia de 3,3 V a generatiilor anterioare, bufferele de I/E au fost proiectate astfel încât sa permita ambele nivele ale semnalelor. |

|  |
| --- |
| Introducerea unei versiuni de 16 biti a magistralei a fost destinata aplicatiilor la care este importanta reducerea numarului de pini. Aceasta versiune poate fi implementata fie prin crearea unei magistrale independente, fie prin divizarea unei magistrale de 64 de biti în 4 segmente decâte 16 biti. |

Începând cu anul 1992, **VESA Local bus** (VL-Bus) a fost cel mai raspândit model de magistrala locala. VESA reprezenta o organizatie non-profit fondata de NEC.

          Magistrala VL-Bus poate transfera 32 de biti simultan si deci permite ca transferul datelor între CPU si un subsistem compatibil video sau un hard-disc sa se faca pe lungimea integrala de 32 de biti a cuvântului. Din punct de vedere fizic, conectorul VL-Bus este o extensie a sloturilor de baza folosite în orice tip de sistem. De exemplu daca avem un sistem ISA, conectorul VL-Bus este montat ca o extensie a sloturilor ISA pe 16 biti, iar daca avem un sistem EISA sau MCA, conectorii VL-Bus sunt extensii ale sloturilor existente.

**Magistrala AGP**(Accelerated Graphics Port) este o magistrala dedicata redarii imaginilor video si grafice de înalta calitate care are la baza modelul PCI, dar beneficiaza de câteva îmbunatatiri suplimentare.

          AGP este o conexiune de mare viteza, care functioneaza la o frecventa de baza de 66MHz, cu latimea de banda de 32 de biti si care ajunge sa faca patru transferuri de date pe un ciclu.

          Intel a conceput magistrala AGP nu numai pentru performante video mai bune, dar si pentru a realiza o legatura directa de mare viteza între placa video si memoria RAM de sistem, reducând astfel nevoia de tot mai multa memorie video.

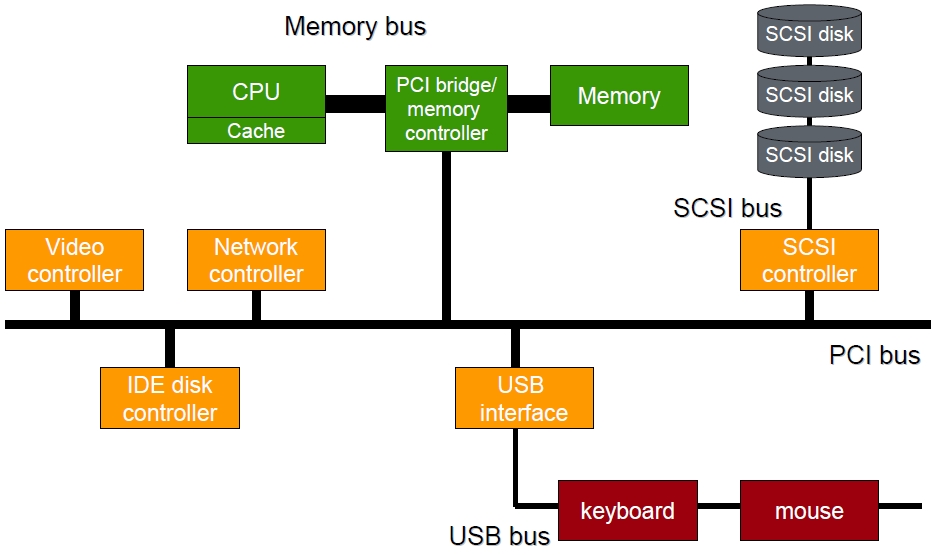
1. **Controlere specializate**

Un sistem de calcul de uz general este constituit dintr-un procesor și un număr de controlere de dispozitive care sunt conectate printr-o magistrală comună. Un controler este o interfață a dispozitivului periferic vazută de sistemul de operare . Programarea dispozitivului se realizează prin intermediul controler-ului asociat. Rolul acestuia este de a transforma comanda primită de la procesor în informatțe specifică dispozitivului.

Un controler deține un set de registre utilizate pentru comunicația cu procesorul. Prin scrierea în aceste registre sistemul de operare poate comanda dispozitivul să transmită informație, să citească informație, sau orice altă acțiune.

Tipuri de controlere pot fi : SCSI, RAID, keyboard controller, disk controller, graphics controller.

Un controler are un buffer de stocare tampon local și un set de registre cu destinație speciala. Controlerul este responsabil pentru transferul datelor între dispozitivele periferice, controlând și bufferul local. Dimensiunea buffer-ului local diferă de la un controler la altul, în funcție de dispozitivul controlat.



Principalele tipuri de controlere utilizate in PC-urile actuale

Link: <http://www.viaclab.utcluj.ro/Curs_1_oct2009.pdf>

Magistralele au rolul de a conecta microprocesorul la memorie sau la adaptoarele care fac posibilă atașarea altor dispozitive prin porturile sau conectorii lor de extensie.

Setul de adaptoare permit microprocesorului să controleze şi să comunice cu echipamentele I/O şi de stocare. Aceste adaptoare sunt ansambluri de circuite care se ataşeaza magistralelor sistemului si care convertesc fiecare magistrala într-un port de interfaţa care acceptă conectarea anumitor echipamente I/O. De exemplu, un adaptor de port serial se conectează la o magistrală a calculatorului şi se creaza un port COM ( de comunicare ) la care poate fi conectat un modem. Adaptorul este deci puntea de legatură între magistralele calculatorului şi dispozitivele care trebuie conectate la el.

Dispozitivele periferice se conecteaza la magistrala prin intermediul unor interfețe care se mai numesc si controlere, adaptoare, drivere, care au rolul:

* de a controla traficul intre periferie si magistrala;
* de a transforma semnalele din serie in paralel sau invers, realizand compatibilitatea intre emitatorul si receptorul semnalului;
* de a converti semnalele care au codificari diferite;
* de a pregati semnalul pentru teletransmitere;

Controlere moderne de disc sunt integrate în unitatea de disc. De exemplu, discuri numite"discuri SCSI" au încorporate controlere SCSI.

Cele mai frecvente tipuri de interfețe oferite în prezent de către controlerele de disc sunt PATA(IDE) si Serial ATA pentru uz casnic. Discurile de ultimă generație folosesc SCSI, Fibre Channel sau Serial Attached SCSI. Controlere de disc pot controla, de asemenea, calendarul de acces la memorie flash, care nu este de natură mecanică.

SCSI este mai frecvent utilizat pentru hard disk-uri și unități de bandă, la care se pot conecta o gamă largă de alte dispozitive, inclusiv scanere și unitățile CD, deși nu toate controlerele pot acoperi toate dispozitivele. Standardul SCSI definește seturi de comenzi pentru anumite tipuri de dispozitive periferice.

SCSI este disponibil într-o varietate de interfețe. Primul, încă foarte frecvent, a fost SCSI paralel, care folosește un design buss paralel. SPI este înlocuit cu Serial Attached SCSI (SAS), care foloseste un design de serie, dar păstrează alte aspecte ale tehnologiei. Multe alte interfețe care nu se bazează pe standarde complet SCSI pune în aplicare încă protocolul de comandă SCSI.

Network controller.Un controler de rețea (NIC, de asemenea, cunoscut ca un card de interfață de rețea, adaptor de rețea, adaptor LAN, și alți termeni similari) este o componentă hardware a calculatorului care conectează un computer la o rețea de calculatoare sau de internet.

Controlerele de rețea au fost puse în aplicare în mod obișnuit pe carduri de expansiune, care erau conectate la un bus al calculatorului; fiind ieftin implementate conform standardului Ethernet,cele mai multe calculatoare noi au o interfață de rețea încorporata în placa de bază.

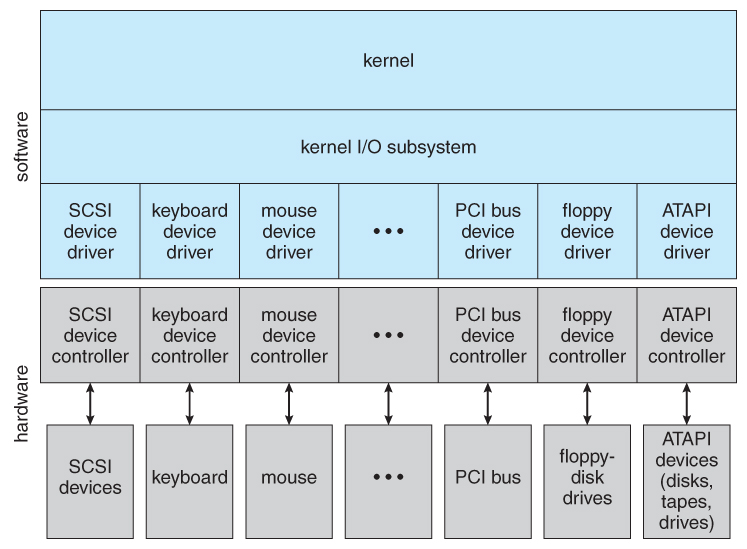
Controlerul de rețea trebuie să comunice prin intermediul circuitelor electronice cu ajutorul unui layer fizic specifici cu link-ul de date standard layer, cum ar fi Ethernet, Wi-Fi sauToken Ring. Aceasta oferă o bază pentru o stiva de protocol de rețea completă, care să permită comunicarea între grupuri mici de computere din aceeași rețea LAN și rețele de comunicații pe scară largă, prin protocoale derutabile, cum ar fiIP.

1. **Drivere de dispozitiv**

Administrarea perifericelor si a echipamentelor se realizează prin intermediul driverelor. Aceste programe sunt autonome de sistemele de operare, de foarte multe ori sunt furnizate de producător echipamentelor și instalate separat. Acestea permit comunicarea dintre sistemele de operare cu echipamentele noi fără a fi modificate, având ca scop interfațarea cu un anumit dispozitiv sau cu o categorie de dispozitive similare.

Comunicarea cu mediul este intermediată de spații tampon, în care datele de intrare acumulate urmează să fie transmise procesorului în ritmul accesibil acestuia, evitând congestia atunci cand apare o aglomerare datorită multitudinii de aplicații.

Aplicatiile utilizator acceseaza o mare varietate de dispozitive prin utilizarea  driverelor de dispozitiv .

  
Structura unui nucleu I/O

Link: http://cursuri.cs.pub.ro/~pso/

**Fluxul unei cereri I/E:**

1. Cererea emite un apel sistem;

2. Kernel-ul verifică parametri, apoi poate returna datele deja buffer-ate și apoi se termină;

3. Cererea este înlocuită din coada“run”,dacă este solicitat un dispozitiv fizic I/O, iar apoi adaugată în coada de așteptare a dispozitivului și apoi este programată cererea I/O;

4. Driver-ul dispozitivului alocă un buffer în kernel și trimite comanda către controler;

5. Controler-ul de dispozitiv hardware-ul pentru a efectua transferul de date;

6. Driver-ul poate aștepta sau lansa o cerere DMA ce va genera întreruperea;

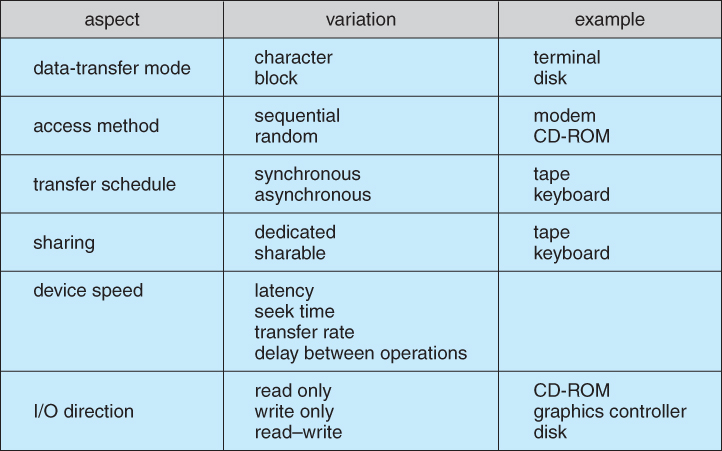
7.Apare întreruperea, handler-ul stochează datele și notifică driverul dispozitivului;

8. Driverul dispozitivului recepționează notificarea și determină starea cererii, iar apoi notifică kernel-ul subsitemului I/O;

9. Kerner-ul transferă datele sau returnează un cod către utilizator și apoi șterge cererea din coada de așteptare;

10.Cererea își reia execuția;

Dispozitivele diferă din punct de vedere al vitezei de transfer, direcţiei de transfer, metodei de acces etc, așa cum se arată în figura următoare:

  
 Caracteristici ale dispozitivelor de I / O.

Link:http://cursuri.cs.pub.ro/~pso/

1. **Definirea canalelor I/O**

Prin intermediul procesorului auxiliar care controlează operatiile cu perifericele se permite cuplarea, indirectă, la unitatea centrală, a mai multor dispozitive periferice. Această unitate funcţională a calculatoarelor a apărut odată cu generatia a doua de calculatoare sub numele de canal de intrare-iesire*.*

Evoluţia în timp a canalelor I/O este în acelaşi timp o evoluţie a creşterii complexităţii si performanţelor.

Pot fi enumerate următoarele etape:

* + CPU controlează direct echipamentele periferice.
  + Este adăugat un modul I/O (o interfaţa serială sau paralelă, programabilă).CPU comandă EP prin transfer programat (direct sau prin interogare)
  + Aceeaşi configuraţie , dar transferul are loc prin înteruperi. Modulul I/O are acces direct la memorie prin DMA. Modulul poate muta informaţia direct în memorie, accesul la CPU fiind necesar doar la începutul si sfârşitul transferului.
  + Modulul I/O foloseşte un microcontroler cu instrucţiuni proprii. CPU programează procesorul I/O pentru un transfer, pe care acesta îl execută folosind instrucţiunile proprii. Când transferul se termină, procesorul I/O întrerupe CPU pentru a comunica rezultatele transferului.
  + Microcontrolerul are memorie locală. El poate controla astfel mai multe EP cu o intervenţie minima din partea CPU. Memoria locala poate fi folosită şi ca buffer de date realizând astfel o rată de transfer mare.
  + Evoluţia microcontrolerelor integrate a fost paralelă cu cea a procesoarelor. Dacă la procesoare s-a urmărit o creştere a vitezei de prelucrare, prin creşterea tactului, creşterea memoriei cache, lărgirea bus-uli de date si adrese, la microcontrolere s-a urmărit integrarea de cât mai multe subsisteme utile (memorie EPROM, RAM,convertoare A/D si D/A).

Conexiunea între un modul I/O si unitatea centrală poate fi:

* Punct la punct : linii dedicate pentru transferul de date (tastatura, mouse-ul)
* Multipunct : la liniile de interfaţa se pot lega mai multe EP (module I/O ca la SCSI)

**8. Concluzii:**

Rolul dispozitivelor de intrare - iesire este acel de a asigura comunicarea între unitatea centrală de prelucrare şi mediul exterior prin intermediul unei unităţi de interfaţă.

Principalele funcţii ale dispozitivelor de intrare- iesire, în cadrul unui sistem de calcul pot fi grupate astfel:

* asigură afişarea/tipărirea rezultatelor  prelucrării într.o formă accesibilă utilizatorului;
* permite utilizatorului posibilitatea de a supraveghea şi interveni, pentru asigurarea funţionării corecte a sistemului, în timpul unei sesiuni de lucru;

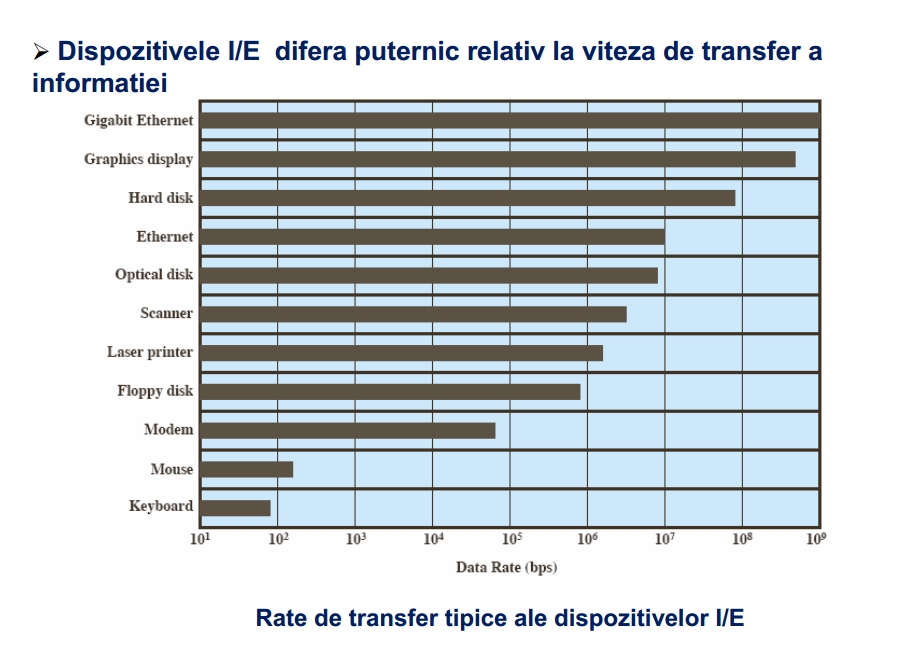
Diferenţa dintre magistrala si port este aceea că: în timp ce magistrala este destinată comunicării mai multor componente între ele, care o împart, pe un port comunică doar două.

Viteza de lucru a perifericelor diferă în funcţie de performanţele lor tehnologice (foarte mică la dispozitivele exclusiv mecanice), dar este mult mai mică decât a unităţii centrale, care funcţionează pe principii exclusiv electronice

Datorită controlerelor specializate a crescut puterea de calcul a unităţii centrale, deoarece acestea au preluat din sarcina unităţii centrale.

Au dispărut modem-urile ce au avut viteze de zeci de kbps şi au apărut, în schimb, plăcile de reţea cu viteze mult mai mari, de ordinul Gbps.

Din punct de vedere al stocării datelor, nu se mai folosesc floppy-urile, ci stick-urile care permit stocare de ordinul zecilor de Mb.



1. **Bibliografie:**

* <http://elearning.masterprof.ro/lectiile/informatica/lectie_01/dispozitive_periferice_de_intrare.html>
* <http://www.koyos.ro/imprimare-si-copiere/multifunctionale/multifunctionale-laser.html>
* <https://www.csensors.com/ccd_page.html>
* <http://lerablog.org/technology/hardware/computers/touchscreen-devices-improve-hospital-care-for-patients/>
* <http://imprimante.pricer.ro/preturi/hewlett-packard-imprimanta-cu-cerneala-hp-deskjet-2000-ch390b>
* Abraham Silberschatz, Greg Gagne, și Peter Baer Galvin, "Concepte ale sistemului de operare, Editia 8", capitolul 13
* <http://ora-tic.wikispaces.com/07.+Sistemul+de+intrare+-+iesire>
* <http://www.dbasquare.com/2012/04/18/analyzing-io-performance/>
* <http://vega.unitbv.ro/~romanca/AOC/Cap7-AOC-IntrareIesire.pdf>
* <http://www.syscom18.info/Articol.asp?ID=189>
* http://www.cs.ucv.ro/staff/cpatrascu/SIE/Structura%20sistemului%20IE.pdf

Louise Muhlbacher

**Introducere**

In ceea ce priveste conectarea placilor de extensie-periferice intr-unul dintre sloturile libere de pe placa de baza,simpla cuplare fizica a acestora e suficienta din punct de vedere hardware, daca dispozitivele sunt de tipul plug’n’play.

Exista si o integrare de tip software,pe baza driver-elor care insotesc,de obicei placile de extensie si,de asemenea,este posibila partial configurarea pe baza setarilor din CMOS, solutii care insa nu rezolva integral problema interconectarii.

Pentru a integra complet un dispozitiv periferic intr-un sistem de calcul,indiferent de rolul acestuia,este nevoie de a defini și a asigura comunicatia generala ,prin care se va semaforiza și controla,la comanda microprocesorului,activitatea totala a dispozitivului în cauza,respective,de a-l cupla in sistemul general al transferurilor de date.

In continuare voi vorbi despre implicarea sistemului de operare(acțiuni software) in ceea ce privește gestiunea de intrari și iesiri:

**Concepte.Probleme**

Exista 3 concepte importante.

Un concept important in design SO este cunoscut sub numele de “independenta dispozitivelor”. Ceea ce inseamna ca este este posibil sa se scrie un program care sa poata sa acceseze orice dispozitiv I/O fara sa fiu nevoita sa-i specific dispozitivul in avans.

Un astfel de exemplu este acela al programului de citire al unui fisier ca intrare indiferent daca se gaseste pe floppy disk,CD,hard disk fara sa fiu nevoita sa modific programul pt fiecare dispozitiv in parte.

De asemenea este posibil sa scriu urmatoarea comanda:

sort < input > output

si aceasta sa functioneze indiferent daca la intrare este vorba de floppy disk, IDD,SCSI,disk iar la iesire avem orice mediu de stocare or monitor.Aceasta problema intra in responsabilitatea SO,care trebuie sa rezolve problema cauzata de faptul ca aceste dispozitive sunt diferite si necesita comenzii diferite sa scrie si sa citeasca.

Foarte apropiat de acest concept este si scopul uniformizarii numelor.Numele unui fisier trebuie sa fie intreg si sa nu depinda in nici un fel de dispozitiv.In UNIX toate discurile pot fi incluse intr-un sistem de ierarhizare in moduri arbitrare astfel incat utilizatorul sa nu aiba nici o grija referitoare daca numele corespunde unui anumit dispozitiv.

De exemplu floppy disk poate fi fixat in fata directoarelor /usr/ast/backup astfel daca copiez un fisier la /usr/ast/backup/Monday sa copiez fisierul pe floppy disk.In felul acesta toate fisierele sunt indexate in acelasi fel: dupa o cale precis definita.

O alta problema problema importanta pt S/O este buna organizare a erorilor.In general erorile trebuie sa fie cat mai bine organizate si apropiate de hardware.Daca controlerul descopera o eroare de citire,ar trebui sa incerce sa corecteze singur eroarea daca poate.Daca nu poate atunci driverul dispozitivului ar trebui sa incerce s-o repare,probabil prin incercarea de a citi din nou adresa de date.Multe erori sunt trecatoare,un astfel de exemplu este eroarea de citire cauzata de praful de pe capul de citire,care va disparea daca operatia de citire se va repeta de cateva ori.Numai daca etajele inferioare nu pot sa solutioneze problema,atunci trebuie sa se apeleze la cele superioare.De cele mai multe ori,recuperarea erorilor poate fi facuta transparent la un nivel scazut fara ca etajele superioare sa stie de eroarea respectiva.

O alta problema este blocarea versus intreruperea.Majoritatea dispozitivelor I/O sunt cu intrerupere - procesorul porneste transferul si inceteaza sa faca altceva pana cand intreruperea soseste.Programele utilizatorului sunt mai usor descris daca operatiile I/O sunt blocate - operatia de citire e suspendata pana cand datele sunt valabile in buffer.

Alta problema o reprezinta memorarea intermediara a datelor.De obicei datele care provin de la un dispozitiv nu pot fi stocate direct in locatia lor finala.De exemplu,cand se primeste un pachet de pe retea ,SO nu stie unde sa-l puna pana nu-l stocheaza undeva si il analizeaza.Memorarea intermediara a datelor implica o serie considerabila de copieri , fapt ce are un impact major asupra performantelor I/O.

Un ultim concept prezentat este acela al dispezitivelor dedicate vs cele generalizate.Unele dispozitive ,cum ar fi discurile,pot fi utilizate de mai multi utilizatori in acelasi timp.Alte dispozitive cum ar fi discurile cu banda magnetica sunt dedicate unui singur utilizator.Introducerea dispozitivelor dedicate a adus si la aparitia unor probleme ,gen deadlocks.

Inca o data SO trebuie sa se descurce atat cu cele dedicate ,cat si cele generale fara sa apara erori.

**Cum lucreaza I/O**

Precum spuneam si in introducere, pentru a integra un dispozitiv periferic trebuie sa stabilim si sa rezervam un canal de comunicatie DMA(Direct Memory Acces),canale de intrerupere IRQ(Interrupt Request) si la final sa setam adresele de intrare-iesire (adrese I/O),la nivelul carora se efectueaza transmiterea datelor,comenzilor si parametrilor necesari proceselor executate de periferic. Astfel, exista 3 cai fundamentale (dpdv soft) diferite prin care I/O lucreaza:

1.I/O programate;

2.I/O intrerupte;

3.I/O care foloseste DMA;

**1.I/O programate**

Cea mai simpla modalitate I/O este sa aiba un processor care sa faca toata treaba pt el;aceasta metoda se numeste I/O programate.

Cel mai simplu mod de a pune in evidentza este sa folosim un exemplu.Consideram ca utilizatorul doreste prin intermediul unui program sa printeze 8 caractere dintr-un sir “ABCDEFGH” cu ajutorul imprimantei.Prima data se pune sirul in memoria intermediara din spatial utilizatorului.Dupa aceea procesul utilizarorului face un apel catre imprimanta pt ca aceasta sa inceapa scrierea.Daca imprimanta este deja folosita de catre un alt proces,atunci aceasta incercare esueaza si se emite o eroarea ori se asteapta pana cand imprimanta este disponibila,aceasta decizie depinde de sistemul de operare si de parametri procesului .De indata ce se elibereaza imprimanta,procesul utilizatorului face un apel catre sistemul de operare prin care ii cere sa inceapa printarea sirului de caractere cu ajutorul imprimantei.Sistemul de operare atunci de obicei copiaza din memoria intermediara sirul intr-o zona,sa zicem p,in nucleu principal de unde se poate accesa cel mai usor.Atunci se verifica daca imprimanta este disponibila,daca nu se asteapta pana aceasta este.De indata ce se intampla asta,sistemul de operare copiaza primul caracter in registrul de date al imprimantei,in acest caz folosind memorie indexata I/O.Aceasta actiune activeaza imprimanta.Se poate ca ,caracterul sa nu apara deoarece unele imprimante asteapta sa se copieze in memoria intermediara mai multe caractere chiar si o pagina pana sa inceapa printarea.De indata ce se copiaza primul caracter la imprimanta,sistemul de operare verifica daca imprimanta este pregatita sa primeasca alt caracter.In general imprimanta are un registru in care se specifica starea ei.Scrierea de date in registru face ca starea imprimantei sa fie ocupata.Cum controlerul imprimantei a scris caracterul current,atunci starea imprimantei se schimba prin trimiterea uni bit in registrul sau ori prin scrierea unei valori in el.In acest punct sistemul de operare asteapta iar ca imprimanta sa fie disponibila.Acest procedeu continua pana cand se printeaza intreg sirul de caractere.

Aceste actiuni prezentate mai sus sunt sintetizate in urmatoarele linii de cod:

copy\_from\_user(buffer, p, count); /\* p is the kernel buffer \*/

for (i = 0; i < count; i++) { /\* loop on every character \*/

while (\*printer\_status\_reg != READY) ; /\* loop until ready \*/

\*printer\_data\_register = p[i]; /\* output one character \*/

}

return\_to\_user();

**Avantaje/Dezavantaje:**

I/O programate sunt simple dar au dezavantajul de a tine CPU ocupat pana cand se termina toate operatiile de I/O.Daca timpul alocat pt printarea unui caracter este scurt atunci asteptarea cat este ocupat este buna.

**Concluzie:**

De obicei asteptarea este mare de aceea este nevoie de o alta metoda I/O.

**2.I/O intrerupte**

Acum vom analiza cazul in care imprimanta nu copiaza caracterele in nici o memorie intermediara ci le printeaza asa cum vin.Daca imprimanta poate sa printeze sa zicem 100caractere/secunda atunci fiecarui character ii trebuie 10ms sa fie printat.Asta inseamna ca dupa fiecare scriere a unui caracter in registrul imprimantei, SPU intra in asteptare pt 10ms asteptand urmatorul caracter sa fie scos.Acest este un timp mai mult decat suficiet ca procesorul sa castige timp si pt alte procese care altfel s-ar fi risipit.

Felul prin care reusim sa facem procesorul sa execute si alte operatii pana imprimanta este pregatita pt a fi folosita este acela de a utiliza intreruperile.Cand sistemul apeleaza imprimanta sirul de caractere este deja facut,memoria intermediara este copiata in nucleul central,cum am aratat mai devreme si primul caracter este copiat la imprimanta de indata ce este disponibila aceasta.In acest punct procesorul executa alte procese.Procesul care se ocupa de printare este blocat pana cand nu ne printeaza intreg sirul.

Acest lucru este facut de urmatoarele instructiuni:

Copy\_from\_user(buffer, p, count);

enable\_interrupts();

while(\*printer\_status\_reg != READY) ;

\*printer\_data\_register = p[0];

scheduler();

Cand imprimanta a printat un caracter si este pregatita sa-l accepte pe urmatorul este generata o intrerupere.Acesta intrerupere opreste aceast proces si ii salveaza stare lui.Atunci se ruleaza procedura de intrerupere la imprimanta care este prezentata intr-o varianta in liniile urmatoare :

if(count == 0) { unblock\_user();

} else {

\*printer\_data\_register = p[i];

count = count – 1;

i = i + 1;

}

acknowledge\_interrupt();

return\_from\_interrupt();

**3.I/O care foloseste DMA(Direct Memory Acces)**

Un aspect deranjant la intreruperea I/O e ca intreruperea intervine la fiecare caracter.Intreruperea ia timp ,fapt ce face ca acest lucru sa ocupe mult timp alocat CPU.O solutionare e sa folosirea DMA.

In esenta, cand este vorba despre DMA, se face practice referire la transferurile de date (si la controlul acestora) intre memorie si porturile I/O, procesorul nemaiavand control asupra lor.

Astfel,aceasta cale consta in trimiterea de caracterele catre imprimanta fara ca procesorul sa fie deranjat.

In esentza DMA este I/O programate,cu diferenta ca toata treaba este facuta de controlerul DMA nu de processor.

Un exemplu de cod de prinatare a unui sir folosind DMA este urmatorul:

copy\_from\_user(buffer, p, count);

set\_up\_DMA\_controller();

scheduler();

**Avantaje/Dezavantaje:**

Marele castig al folosirii DMA este acela ca se reduce numarul intreruperilor de la unul la un caracter la unul pe memoria intermediara de printare.Daca avem multe caractere si intreruperile incite,acest lucru este o imbunatatire mare.Pe de alta parte ,controlerul DMA este mult mai lent decat procesorul.

**Concluzie:**

Daca controlerul DMA nu este folosit la capacitatea lui maxima si processor nu are altceva de facut in timp intreruperii DMA,atunci I/O programate si I/O intrerupte pot fi mai bune.

Bibliografie:

1. Andrew S. Tanenbaum “Modern Operating Systems”