

Gestiunea perifericelor

Tipuri de clasificări:

- **periferice de intrare:** tastatura
- **periferice de ieșire:** ecranul
- **periferice bidirecționale:** extensiile de memorare, HDD, Floppy

În funcție de cui i se adresează

- **human readable** (ieșire și intrare de la operator uman: tastatură, ecran)
- **machine readable** (HDD, modem)

Din punct de vedere al funcționalității

- **de tip caracter** (intrare + ieșire: octet cu octet; tastatură, imprimantă, ecran)
- **de tip bloc** (informația se citește în blocuri cu o dimensiune dată de fiecare sistem: discul)

I. Gestiunea perifericelor de tip caracter

Toate perifericele posedă un port de date care este de tip *in* sau *out* și un port de stare numai *in* – putem citi starea perifericului.

ROL: La perifericele de intrare în informația care cuprinde starea perifericului avem o informație de tip *boolean*:

ex.: RXRDY = 1 (este ceva pe „țeavă”, are sens să citesc portul de date)

La perifericele de ieșire: TXRDY

- stare normală = 1 (perifericul e gata să primească date)

Filisofii de a scrie drivere pe perifericele

1. Polling (veche)

Intrare:

1. RXRDY?
2. NU – mergi la 1
3. Citește caracter

Ieșire:

1. TXRDY?
2. NU – mergi la 1
3. Transmite caracter

2. Pe întreruperi (modernă)

Fiecare periferic poate să fie legat (hardware) la un nivel de întrerupere - se produce când RXRDY (respectiv TXRDY) trece din 0 în 1 (era ocupat și nu mai este).

Intrare: (Programul principal – consumatorul – și întreruperea comunică printr-un buffer circular)

1. Buffer gol?
2. NU – returnează caracter
3. DA – return

Întreruperea (producătorul):

1. Citește caracter
2. Depune în buffer

Ieșire:

1. Buffer plin?
2. NU – depune în buffer
3. DA - return

NU

DA

Ieșire:

1. TXRDY?
2. DA – transmite caracter, return
3. Buffer plin?
4. NU – depune în buffer
5. DA - return

Întreruperea:

1. Buffer gol – return
2. Citește caracter din buffer
3. Transmite caracter

La ieșire programul principal este producătorul iar întreruperea consumatorul.
Principiile saltului de IO: întreruperea se realizează la trecerea din 0 în 1.

II. Gestiunea perifericelor de tip bloc

Perifericele de tip bloc corespund în general perifericelor care reprezintă extensia memoriei (HDD).

Observatie: În UNIX blocurile sunt de 512 octeți; marcați în memoria cache

SCOP: minimizarea numărului de accese la disc

fflush (FILE *fp) forțează golirea cache-ului chiar dacă nu era plin

Există trei modalități de golire a bufferului:

- a. Bloc clasic (implicit)
- b. Golire la newline
- c. Autoflush (cum am scris un caracter, cum este pus pe disc)

Observatie: În UNIX perifericele sunt cazuri particulare de fișiere.

Discurile sunt un periferic de tip special alcătuit din cilindrii, piste și sectoare.

Operații:

1. Poziționare cap R/W (penalizantă, deoarece este mecanică, deci trebuie minimizat numărul de mișcări al capului de R/W);
2. Rotire
3. Citire/Scriere propriu-zisă

Politici posibile:

1. *FIFO* (nu este bună, dar anumite periferice (precum imprimanta) nu pot fi tratate decât așa)
2. *Cea mai apropiată cerere față de poziția curentă* (nu este bună, deoarece este inechitabilă)
3. *Tehnica ascensorului*

date de intrare: poziția curentă și direcția de deplasare (0 – jos, 1 – sus)
următoarea cerere care va fi servită este cea mai aproape de poziția curentă, dar în direcția de deplasare; dacă pe direcția de parcurgere nu mai găsește nicio cerere, schimbă direcția.

Gestiunea fisierelor

a

Un fișier este teoretic o informație (un șir mare de octeți) care este alcătuită din înregistrări, iar înregistrările sunt alcătuite din câmpuri.

O colecție de fișiere formează o bază de date: flat file.

Din punct de vedere al **structurii**, fișierele pot avea:

- **format fix:** fiecare înregistrare același număr de octeți, fiecare câmp același număr de octeți.
- **format variabil:** înregistrările nu au aceeași lungime; nu au același număr de înregistrări.
ex.: fișier text: înregistrări = linii; câmpuri = cuvintele de pe linii;

Noțiuni generale despre fișiere

Operații asupra fișierelor

1. deschidere
2. exploatare
3. închidere

Mai amănunțit:

1. deschidere
2. închidere
3. citire
4. modificare (suprascriere)
5. scriere la sfârșit
6. ștergere informație
7. ștergere fișier
8. redenumire fișier
9. copiere fișier
10. modificare caracteristici

Caracteristici

1. Proprietarii
2. Drepturi

La deschiderea fișierului există o informație: „poziție curentă”. (la deschidere: poziția curentă este pe primul octet; citirile și scrierile modifică poziția curentă spre sfârșitul fișierului)

Modalități de acces

1. **secvențial** (dacă vreau să citesc octetul n trebuie să citesc $n-1$ octeți dinainte)
2. **aleator** (permite să modific poziția curentă fără să fac operații de citire/scriere și în aval și în amonte)

Observație: În C : cu ajutorul apelului **fseek**

3. **secvențial indexat** (nu este build-in în sistemul de operare, depind de anumite softuri, biblioteci)