

# Operációs rendszerek 1. – 6. előadás

## A virtuális gép hardvereszközei

Soós Sándor

Nyugat-magyarországi Egyetem  
Simonyi Károly Műszaki, Faanyagtudományi és Művészeti Kar  
Informatikai és Gazdasági Intézet

E-mail: [soossandor@inf.nyme.hu](mailto:soossandor@inf.nyme.hu)



# Tartalomjegyzék

## 1 Ismétlés

- Emlékeztető az előző óráról

## 2 A virtuális gép eszközei

- Táruk
- Tárkezelés az operációs rendszerben
- Fájlkezelés az operációs rendszerben

## 3 Befejezés

- Emlékeztető kérdések



# Hol tartunk?

## 1 Ismétlés

- Emlékeztető az előző óráról

## 2 A virtuális gép eszközei

- Táruk
- Tárkezelés az operációs rendszerben
- Fájlkezelés az operációs rendszerben

## 3 Befejezés

- Emlékeztető kérdések



# Klasszikus konkurens problémák

- Termelő-fogyasztó probléma
- Írók-olvasók problémája
- Étkező filozófusok problémája
- Adatfolyamok illesztése



# Nyelvi eszközök a folyamatok programozására

- Precedenciagráf
- Fork-join utasításpár
- Parbegin-parend utasításpár



Ismétlés vége



# Hol tartunk?

## 1 Ismétlés

- Emlékeztető az előző óráról

## 2 A virtuális gép eszközei

- Tárák
- Tárkezelés az operációs rendszerben
- Fájlkezelés az operációs rendszerben

## 3 Befejezés

- Emlékeztető kérdések



# A virtuális gép további eszközei

- Mielőtt elkezdjük:
  - FIGYELEM! Jegyzetelés!!!
  - Jövő héten zárthelyi!
  - Legyen-e konzultáció? Mikor?
- Az elmúlt órákon megismerkedtünk a virtuális gépeken futó folyamatokkal és az azokat futtató processzorokkal
- Most megvizsgáljuk, hogyan kezeli az operációs rendszer a virtuális gép további eszközeit
- Kezdjük azzal, hogy hogyan tároljuk az adatokat!





# Táruk, tárhierarchia, I

- A táruk hierarchikus rendbe szervezettek:

külső táruk, harmadlagos tárolók
háttértáruk, másodlagos tárolók
operatív tár, memória
a processzor regiszterei

- A táruk jellemzői hierarchia szintek szerint:
  - Minél magasabb szinten van egy tároló:
    - annál nagyobb méretű
    - annál lassabb működésű
    - annál nagyobb egységekben címezhető
    - annál hosszabb a tárolási idő



# Táruk, tárhierarchia, II

- Alapvető ellentmondás:
  - A különböző tárolási szintek hatékony kezelése a rendszer teljesítményének egyik kulcsa
  - A műveletek elvégzéséhez az adatoknak a processzor regisztereiben kell lenniük. (Miért?)
  - Az összes szükséges adat szinte soha nem fér el a regiszterekben, sokszor a memóriában sem, néha még a háttértárakon sem
  - **A megoldás:** az adatokat rendszeresen mozgatni kell a tárolási szintek között
  - Hogyan?
    - regiszterek  $\leftrightarrow$  memória: processzor
    - memória  $\leftrightarrow$  háttértár: fájlműveletek
    - háttértár  $\leftrightarrow$  külső táruk: felhasználói beavatkozás



## Tárak, tárhierarchia, III

- Adatok elérése, címzés a különböző tárolószinteken:
  - **regiszterek:** minden regiszternek külön neve van, bizonyos műveletek csak bizonyos regiszterekkel végezhetők el
  - **memória:** minden memóriarekesz külön-külön címezhető
  - **háttértár:** fájlanként, azon belül rekordonként, blokkonként címezhető
  - **külső tár:** médiánként címezhető, melyik CD/DVD lemezen, szalagon található a keresett adat
- Az adatok mozgatása kétféleképpen történhet:
  - 1 **Explicit:** (világosan kifejezett) pl. egy utasítással betöltünk egy fájlt a memóriába
  - 2 **Implicit:** (rejtett, közvetett) a rendszer végzi a háttérben a kényelem fokozása, vagy a hatékonyság növelése érdekében



# Tárak, tárhierarchia, IV

- A rejtett adatmozgatás tipikus fajtái:

## 1 Virtualizálás

- Az alacsonyabb szinten lévő tár címezési módját kiterjesztjük a magasabb szintre
- Ezzel megnöveljük az alacsonyabb szintű tár méretét (látszólag), de lassabban működik
- Példa: virtuális memória, lemezen tárolódik, de memória módjára kezeljük, nem fájlként

## 2 Gyorsítótár (cache)

- Magasabb szintű elérési módon kezelünk egy alacsonyabb szintű tárat
- Sokkal gyorsabb
- De a mérete sokkal kisebb, mint a szimulált tár szokásos mérete
- Kulcsfontosságú az adatmozgatás szervezése



# Táruk, tárhierarchia, V

- **Lokalitási elv:** ha egy adatra szükség van, akkor nagy valószínűséggel a környezetében lévő adatokra is szükség lesz
- Ezt használjuk ki a gyorsítótárak adatokkal való feltöltésekor
- Megfelelő adatcserélési algoritmusokkal és a gyorsítótárak megfelelő méretezésével 80 – 99%-os találati arány is elérhető
- Jellegzetes gyorsítótárak:
  - Processzorba épített hardver-gyorsítótárak (utasítás- és adatcache), a memóriában lévő adatok aktuális részét teszik gyorsabban elérhetővé a processzor számára
  - A memóriában kialakított átmeneti tárterületek (buffer-cache) az éppen használatban lévő fájlok adatai egy részének tárolására
  - Memóriában kialakított virtuális diszk (RAM-diszk, elektronikus diszk)
  - A harmadlagos táruk fájlrendszereit tároló mágneslemez területek



# Táruk, tárhierarchia, VI

- Mire kell vigyázni a virtuális táruk kapcsolatban?
  - Mi történik, ha szabálytalanul állítjuk le az operációs rendszert?
  - A memóriában lévő adatok váratlanul elvesznek
  - A rejtett adatmozgatások félbeszakadnak
  - A háttértárukban lévő adatok inkonzisztens állapotban maradnak
  - A mágneslemezeken lévő adatokat nem tudjuk elérni a hagyományos eszközökkel, ha azok adminisztrációja nem hibátlan



# Tárak, tárhierarchia, VII

- Hogyan tudunk védekezni az ilyen hibák ellen?
  - Szünetmentes tápegység, akkumulátoros táplálás, notebook
  - **Vigyázat!** Nem csak áramszünet miatt állhat le szabálytalanul az operációs rendszer!
  - Biztonságos szoftvermegoldások (pl. naplózó fájlrendszer, minden végrehajtott műveletet naplóz a rendszer, így rendszerhiba esetén visszaállítható a korábbi állapot)



# Hol tartunk?

## 1 Ismétlés

- Emlékeztető az előző óráról

## 2 A virtuális gép eszközei

- Táruk
- Tárkezelés az operációs rendszerben
- Fájlkészelés az operációs rendszerben

## 3 Befejezés

- Emlékeztető kérdések





# Tárkezelés az operációs rendszerben

- A tárkezelés területén az operációs rendszer feladata a következő eszközök kezelése:
  - Operatív tár (memória)
  - Háttértárak és külső tárok (fájlrendszerek)
  - Virtuális memória
  - Fájlrendszer-gyorsítótárak (buffer-cache)



# Operatív tár

- A korai operációs rendszerek közvetlenül kezelték a memóriát
- A programok a valóságos fizikai memóriacímekre hivatkoztak
- A program csak egy adott gépkonfiguráción tudott futni
- Javítás: a program betöltésekor a betöltőprogram (loader) beállította a fizikai címeket, ezután indította el a programot
- A mai operációs rendszerekben minden folyamat kap egy úgynevezett logikai memóriát, ami a fizikai memória (operatív tár) egy elválasztott része
- A folyamat nem tudja, hogy a fizikai memóriában ténylegesen hol fut, nem is kell tudnia, sőt ez meg is változhat



# Logikai memória, I

- A megkapott memóriát a folyamat RAM, vagy PRAM-modell szerint működőnek látja
- Ezt a memóriát nem egy egységes, folytonos memóriának kezeli, hanem három önálló részt különít el:

## 1 Kódterület:

- általában csak olvassák a folyamatok
- nem okoz gondot a közös használat
- mérete előre ismert és nem változik

## 2 Adatterület:

- a változókat tárolja
- a folyamatok írják és olvassák
- a mérete általában nem változik



# Logikai memória, II

## 3 Verem (stack):

- változókat tárol
- dinamikusan változik a mérete, nőhet és csökkenhet is
- speciális esetben túlnőhet minden határon (pl. rekurzió), kinőheti a fizikai memóriát
- fel kell készülni a verem-túlcsordulás kezelésére
- (Mit jelent a rekurzió?)
- (Hogyan működik az eljáráshívás?)



# Hol tartunk?

## 1 Ismétlés

- Emlékeztető az előző óráról

## 2 A virtuális gép eszközei

- Tárák
- Tárkezelés az operációs rendszerben
- Fájlkezelés az operációs rendszerben

## 3 Befejezés

- Emlékeztető kérdések



# Háttértárak kezelése

- A memória tartalma addig él, amíg a számítógép működik
- A folyamat szempontjából a memóriában lévő adatok addig élnek, amíg a folyamat fut
- Ha valamilyen adatot meg akar őrizni, akkor háttértárra kell menteni
- A háttértárra írandó adatokat fájlokba kell szervezni
- A felhasználó szempontjából az operációs rendszer legfontosabb feladata a fájlok kezelése (DOS-Disk Operating System)



# Fájlkezelés

- A másodlagos és harmadlagos tárolókon csak fájlokban lehet adatokat tárolni
- A fájlok kezelése az operációs rendszer feladata
- Két szint:
  - ❶ A fájlok, mint tárolási egységek kezelése (egyben)
    - Fájlnev
    - Hierarchikus könyvtárszerkezet
    - Egy, vagy több gyökér (root)
    - Katalógusfájl (directory)
    - Kötetek (volume)
    - Mount
    - A fájl azonosítása: elérési út + fájlnev
  - ❷ A fájlokban lévő adatok kezelése
    - Fájlmodellek
    - Fájlműveletek



# Fájlmodellek

- A fájlban lévő adatok elérésére háromféle fájlmodell használatos:
  - ❶ Soros elérésű (szekvenciális, sequential) fájl
    - mint a mágnesszalag
    - csak sorban lehet írni és olvasni
    - fájlpointer
  - ❷ Közvetlen elérésű (direct) fájl
    - bármelyik adatelem bájt, vagy rekord elérhető a sorszáma alapján
  - ❸ Indexelt, index-szekvenciális elérésű (index sequential access method, ISAM) fájl
    - adatrekordok, adatmezők
    - kulcsmező(k) alapján lehet elérni az adatokat
    - indextábla, indexfájl, rendezett kulcsok, mutató az adatra
    - adatbázis





# Fájlműveletek

- 1 Megnyitás (open)
- 2 Lezárás (close)
- 3 Végrehajtás (execution)
- 4 Létrehozás (create)
- 5 Törlés (delete)
- 6 Adatelérés, írás, olvasás (write, read)
- 7 Hozzáírás, hozzáfűzés (append)
- 8 Pozícionálás (seek)



# Hol tartunk?

## 1 Ismétlés

- Emlékeztető az előző óráról

## 2 A virtuális gép eszközei

- Táruk
- Tárkezelés az operációs rendszerben
- Fájlkezelés az operációs rendszerben

## 3 Befejezés

- Emlékeztető kérdések



## Emlékeztető kérdések

- 1 Hogyan csoportosíthatjuk a számítógépben lévő különböző tárákat?
- 2 Mi jellemzi ezeket a kategóriákat?
- 3 Milyen módokon mozgatjuk az adatokat a tárhierarchia szintjei között?
- 4 Mit nevezünk cache-nek?
- 5 Milyen veszélyei vannak a virtuális tárkezelésnek?
- 6 Hogyan kezelik a memóriát a különböző operációs rendszerek?
- 7 Hogyan kezelik a fájlokat az operációs rendszerek?
- 8 Mutassa be a három jellemző fájlmodellt!
- 9 Hasonlítsa össze ezeket?
- 10 Milyen fájlműveleteket valósítanak meg az operációs rendszerek?



# Befejezés

Köszönöm a figyelmet!

