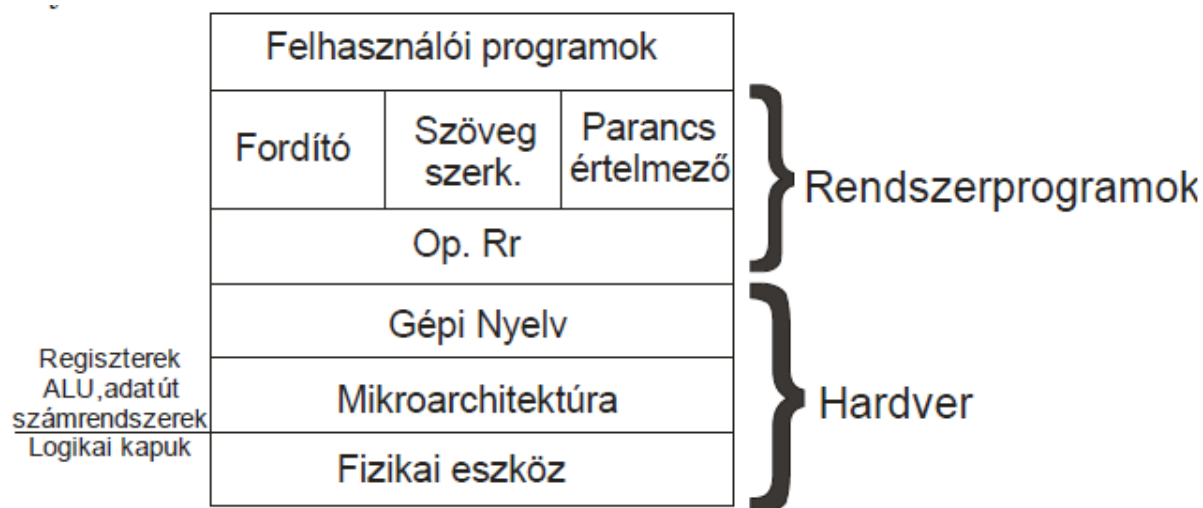


Operációs rendszerek

Absztrakciós szintek

A hardverek fölé egy olyan rendszert helyezünk, amivel a programozó egyszerűbben tudja kezelni az erőforrásokat:



Kernel szerepe, feladatai

Magas szintű függvények, amelyekkel hozzáférünk a hardverhez.

- Eszközkezelő
- Megszakításkezelő
- Rendszerhívás/válasz
- Erőforráskezelés
- Filekezelés
- Memóriakezelés
- Ütemezés
- Holtpontkezelés



Shell szerepe, feladatai

- Alkalmazások futásának kezelése
- Kapcsolat a felhasználóval

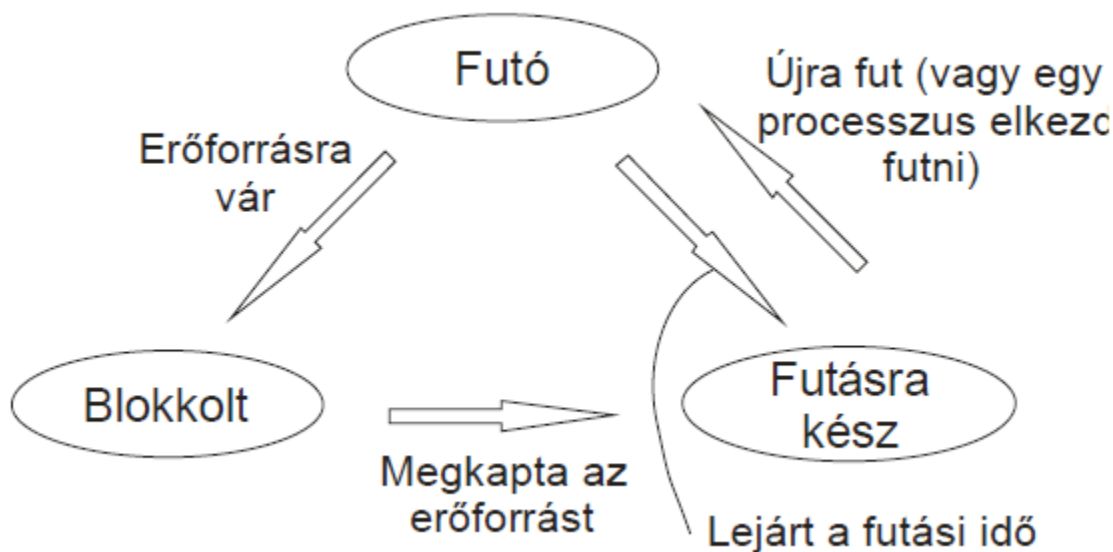
Processzus fogalma és állapotainak ismertetése

Végrehajtás alatt lévő programot jelent. Minden processzus rendelkezik saját címtartománnyal – a memória egy szeletével – amelyen belül olvashat ill. írhat. Minden processzushoz tartozik még egy

regiszterkészlet, beleértve az utasításszámlálót, veremmutatót, egyéb hardver regisztereket, valamint a program futásához szükséges egyéb információt.

Egy processzus három állapotban lehet:

- Futó (az adott pillanatban éppen használja a CPU-t)
- Futáskész (készen áll a futásra; ideiglenesen leállították, hogy egy másik processzus futhasson).
- Blokkolt (bizonyos külső esemény bekövetkezéséig nem képes futni).



Folyamatok ütemezése

Amikor egynél több processzus képes futni, akkor az operációs rendszernek el kell döntenie, hogy melyik fusson először. Az operációs rendszer eme döntéshozó részét, ütemezőnek nevezzük, az erre a célra használt algoritmust pedig ütemezési algoritmusnak.

Ütemezési algoritmusok tulajdonságai

- Pártatlanság – biztosítani, hogy minden processzus megkapja az őt megillető CPU-részt.
- Hatékonyság – a CPU-t 100 százalékig kihasználni
- Válaszidő – minimalizálni az interaktív felhasználók válaszütemét
- Áthaladási idő – minimalizálni azt az időt, amíg a köteget felhasználók eredményre várnak
- Áteresztőképesség – maximalizálni az óránként végrehajtott feladatok számát.

Ütemezési algoritmusok

- Round-Robin
 - legegyszerűbb, legpártatlanabb, széleskörben használt
 - minden processzusnak ki van osztva egy időintervallum amelyet időszeletnek nevezünk, ezalatt az idő alatt engedélyezett a futás

- a futtatandó processzusok egy listában vannak tárolva, ha egy processzus felhasználta az időszeletet a lista végére kerül
- Prioritásos ütemezés
 - minden processzushoz hozzárendelünk egy prioritást és a legmagasabb prioritású processzusnak engedjük meg hogy fusson
- Többszörös sorok
- Legrövidebb feladat először
- Garantált ütemezés
- Sorsjáték ütemezés
- Valós idejű ütemezés
 - Szigorú
 - Lágú

Rendszerhívás fogalma

Az operációs rendszer és a felhasználói programok közötti kapcsolatot az operációs rendszer által biztosított "kiterjesztett utasítás" készlet adja, amelyet hagyományosan rendszerhívásoknak nevezünk.

Többféle rendszerhívásról beszélhetünk:

- Processzuskezelő rendszerhívások
- Szignálkezelő rendszerhívások (váratlan események fellépésekor kapcsolatba lép a processzussal)
- Fájlkezelő rendszerhívások
- A védelem rendszerhívásai (A tulajdonos, a csoport és a többiek jogosultságaival foglalkozik)
- Az időkezelés rendszerhívásai (1970. január 1. 00:00)

Atomi műveletek. Kritikus szekció. Kölcsönös kizárás. Szemaforok.

Atomi művelet

- Olyan művelet, amely nem szakítható meg. Pl: összeadás, szemfar emelése, leengedése

Kölcsönös kizárás

- valamilyen módszer, amely biztosítja, hogy ha egy processzus megosztott fájlt vagy változót használ, a többi processzus tartózkodjon ettől a tevékenységtől.

Kritikus szekció

- A programnak azt a részét, amelyben a megosztott memóriát használjuk, kritikus területnek vagy kritikus szekciónak nevezzük.

Szemafor

- A **szemafor** a számítógép-programozásban használt változó vagy absztrakt adattípus, amit az osztott erőforrásokhoz való hozzáférések szabályozásához használnak a többszálú

környezetekben. Megalkotása Edsger Dijkstra holland matematikusnak, a programozás egyik úttörőjének nevéhez fűződik.

A holtpont fogalma. A holtpont kialakulásának előfeltételei

A folyamatok egy csoportja olyan eseményre vár, amelyet egy másik, ugyancsak várakozó folyamat tud előidézni.

Kialakulásának előfeltételei:

- kölcsönös kizárás (erőforrás használat)
- foglalva várakozás
- nem elvehető erőforrások
- körkörös várakozás.

A virtuális memória. Virtuális memória megvalósítása lapozással.

Tul nagy méretű programok futtatása amely a program rétegekbe van szervezve, ha egy réteg befejeződik hívja a következőt.

A rétegek cseréje OP feladat, de a rétegek darabolása a programozó feladata(mi a megoldás a rétegek közötti adatcserére, alkalmazni a rétegen kívül egy közös részt, ahova elhelyezhetők a rétegek között cserélendő adatok.

A program az adat és a verem együttes területek meghaladhatja a fizikai memória méretét. Az OP csak a program éppen használt részét tartja a memóriába, a program többi része a lemezen van tárolva.

Lapozás

A virtuális memóriát használó rendszerekben leggyakrabban a lapozás technikáját alkalmazzák.

Ezeket a program által generált virtuális címeknek, az egészet pedig virtuális címtérnek nevezik. A virtuális címek nem kerülnek közvetlenül a memóriabuszra, ehelyett a memóriakezelő egységbe (MMU, Memory Management Unit) kerülnek, ami a virtuális címeket képezi le fizikai címekre. A következő ábra erre a leképezésre mutat egy egyszerű példát. A virtuális címek 16 bitesek, 0-tól 64 KB-ig tudnak címezni. A gépnek azonban csak 32 KB memóriája van. A program teljes egészében a lemezen van, és csak azok a részek töltődnek be, amelyekre szükség van. A virtuális címtér lap (page) nevű egységekre osztják, ennek megfelelő egység a fizikai memóriában a lapkeret. A lapok és a lapkeretek mindig pontosan egyforma méretűek.

A laptábla célja az, hogy a virtuális lapokat lapkeretekre képezzük le.

Két probléma:

- a laptábla nagyon nagy lehet
- a leképezésnek gyorsnak kell lennie

Az első pont abból következik, hogy a modern számítógépek legalább 32 bites virtuális címeket használnak. 4 KB-os lapméretnél a címtér több mint egymillió lapból áll. A gyorsaság azért szükséges, mert a leképezést minden memóiahivatkozásnál végre kell hajtani (különben fellép az üvegyakhatás).