

7. tétel

Többfelhasználós operációs rendszerek

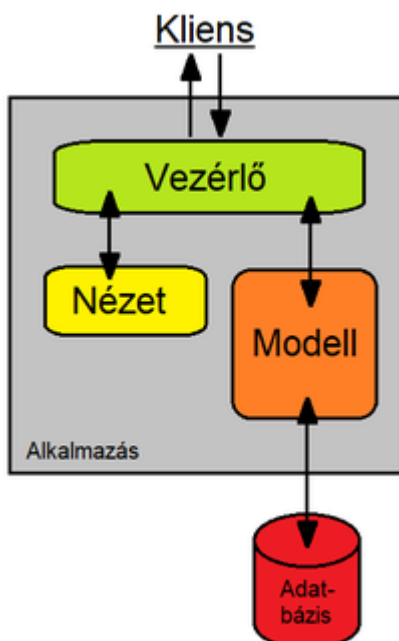
A többfelhasználós operációs rendszerek egyszerre több felhasználónak engednek hozzáférést a számítógéphez. Az időbeosztó rendszereket és a webszervereket is ide sorolhatjuk, hiszen az idő beosztásával több felhasználót engednek ugyanahhoz a számítógéphez. Az egyfelhasználós rendszerek csak egy felhasználót engednek, viszont többféle programot futtathatnak egy időben.

Multi-tasking, single-tasking

A multi-tasking operációs rendszerek az emberi érzékelés szintjén egyszerre több programot képesek futtatni. A single-tasking rendszer egyszerre csak egy programot futtat. A multi-tasking kétféle lehet, vagy preemptív vagy kooperatív. A preemptív multi-tasking rendszer a processzor működését apró időszelvényekre bontja, és ezeket a szeleteket szétosztja a programok között. A kooperatív rendszereknél minden folyamat bizonyos időnként lemond a processzor használatáról, átengedve azt egy másiknak.

MVC (Model-View-Controller)

Az **MVC (Model-View-Controller)**, magyarul **MNV (Modell-Vezérlő-Nézet)** egy szoftvertervezésben használatos programtervezési minta. Összetett, sok adatot a felhasználó elé táró számítógépes alkalmazásokban gyakori fejlesztői kíváncsi az adatahoz (modell) és a felhasználói felülethez (nézet) tartozó dolgok szétválasztása, hogy a felhasználói felület ne befolyásolja az adatkezelést, és az adatok átszervezhetők legyenek a felhasználói felület változtatása nélkül. A modell-nézet-vezérlő ezt úgy éri el, hogy elkülöníti az adatok elérését és az üzleti logikát az adatok megjelenítésétől és a felhasználói interakciótól egy közbejáró összetevő, a vezérlő bevezetésével.



Vezérlő

Az eseményeket, jellemzően felhasználói műveleteket dolgozza fel és válaszol rájuk, illetve a modellben történő változásokat is kiválthat.

Nézet

Megjeleníti a modellt egy megfelelő alakban, mely alkalmas a felhasználói interakcióra, jellemzően egy felhasználói felületi elem képében. Különböző célokra különböző nézetek létezhetnek ugyanahhoz a modellhez.

Modell

Az alkalmazás által kezelt információk tartomány-specifikus ábrázolása.

Az MVC működése:

1. A felhasználó valamilyen hatást gyakorol a felhasználói felületre (pl. megnyom egy gombot).
2. A vezérlő átveszi a bejövő eseményt a felhasználói felületről, gyakran egy bejegyzett eseménykezelő vagy visszahívás útján.
3. A vezérlő kapcsolatot teremt a modellel, esetleg frissíti azt a felhasználó tevékenységének megfelelő módon (pl. a vezérlő frissíti a felhasználó kosarát). Az összetett vezérlőket gyakran alakítják ki az utasítás mintának megfelelően, a műveletek egységbezárásáért és a bővítés egyszerűsítéséért.
4. A nézet (közvetve) a modell alapján megfelelő felhasználói felületet hoz létre (pl. a nézet hozza létre a kosár tartalmát felsoroló képernyőt). A nézet a modellből nyeri az adatait. A modellnek nincs közvetlen tudomása a nézetről.
5. A felhasználói felület újabb eseményre vár, mely az elejéről kezdi a kört.

Virtualizációs szoftverek

A mai modern számítógépek elég nagy teljesítményűek ahhoz, hogy több operációs rendszert futtassanak a fő operációs rendszeren, ami azt jelenti, hogy a virtuális gépek ma sokkal gyakoribbak, mint valaha. A virtualizációs szoftver lehetővé teszi, hogy egy másik operációs rendszerben emulált operációs rendszert futtasson.

A virtualizáció előnyei:

- Helymegtakarítást és üzemeltetési költségeket takarít meg.
- Az adatközpontok egyszerű kezelése.
- Növeli az informatika termelékenységét.
- Segít folytatni a munkát, ha a rendszer összeomlik vagy hirtelen hibákat észlel.
- Az alkalmazások és az erőforrások gyorsabban elérhetők a virtualizáció használata közben.



Robusztusság

Robusztus egy alkalmazás, ha megakadályozza vagy futás közben kiszűri a programozási hibákat. A robusztusságot nyelvi szinten a szigorú, statikus típusosság biztosítja. Minden adatnak fordításkor jól definiált típusa van, nincsenek automatikus konverziók, az explicit konverzió csak kompatibilis típusoknál sikerül, egyébként legkésőbb futtatáskor programhibát okoz. A mutatók megszüntetésével rengeteg potenciális hibalehetőség tűnt el a nyelvből, bár ennek nem minden programozó örül. A dinamikus szemétygyűjtés megkímél bennünket a hasznos memória elszivárgásától.

Csúcsterhelés

Valaminek az átlagos terhelését meghaladó, legnagyobb terhelés.