5. előadás

Folyamatok

2. rész

Elnevezési rendszerek

1. rész

Folyamatok

2. rész

Emlékeztető

- Folyamatok
- Szálak
- Könnyűsúlyú folyamatok

- Szálak nem elosztott rendszerekben
  - rendszerhívásra várakozás alatt hasznos munka
  - többprocesszoros rendszerek kihasználása
  - implementációs okok

Többszálú kliens

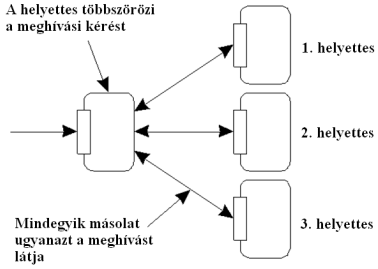
- Előnyök:
  - elosztottsági átlátszóság
  - hatékonyság

- Kliens (tipikus) feladatai:
  - felhasználói felület
  - helyi feldolgozó rész
  - kommunikációs rész
  - …
  - + elosztottsági átlátszóság eléréséhez szükséges elemek

Kliensoldali szoftver és az átlátszóság

- Hozzáférhetőségi átlátszóság
- Elhelyezkedési, áthelyezhetőségi és mozgathatsági átlátszóság
- Meghibásodási átlátszóság
- Többszörözhetőségi átlátszóság

Többszörözhetőségi átlátszóság megvalósítása kliens-oldalon

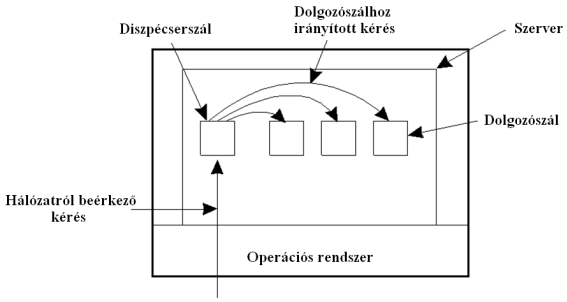


A távoli objektumtöbbszörözés átlátszóvá tételének egyik lehetséges módja a kliensoldali megoldás használata.

Többszálú szerver (1)

- Előnyök:
  - egyszerűbb kód
  - párhuzamos végrehajtás lehetősége
  - blokkoló rendszerhívás idejének kihasználása
- Szerver feladatai:
  - várakozik a beérkező kérésekre
  - kezeli a kérést

Többszálú szerver (2)

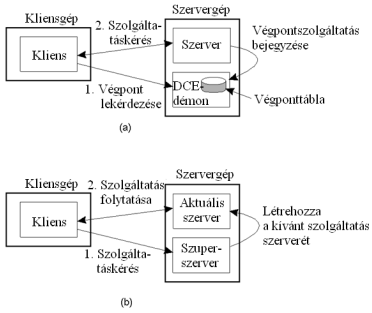


A diszpécser/dolgozó modell szerinti többszálú szerver felépítése.

Többszálú szerver – általános tervezési szempontok

- Ismétlőszerver <-> konkurens szerver
- Lehet-e megszakítást küldeni a szervernek
- Állapot nélküli szerver <-> állapotteljes
- Hogyan találják meg a kliensek a szervert

Hogyan találják meg a kliensek a szervert

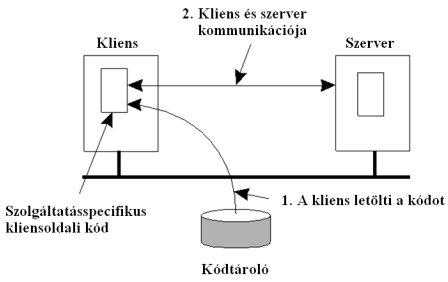


a) démon használatával (mint a DCE esetén szokásos)
b) szuperszerver használatával (mint a Unix-rendszerekben szokásos)

Kód áthelyezése

- A kód áthelyezésének okai:
  - teljesítménynövelés – terhelésmegosztási algoritmusok
  - kommunikáció leszorítása – adatot ott dolgozzuk fel, ahol van
  - párhuzamos végrehajtás kihasználása
  - rugalmasság

A kód áthelyezésének okai



A kliens szerverrel folytatott kommunikációjának dinamikus konfigurálási elve. A kliens először letölti a szükséges szoftvert, majd meghívja a szervert.

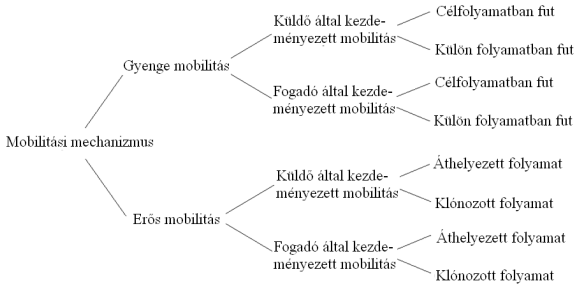
A folyamat szegmensei

- Kódszegmens – végrehajtott program
- Erőforrásszegmens – igényelt külső erőforrás-hivatkozások
- Végrehajtási szegmens – futási állapot

Kódáthelyezési modellek (1)

- Gyenge mobilitás
- Erős mobilitás
- Küldő által kezdeményezett
- Fogadó által kezdeményezett
- Gyenge mobilitásnál
  - Célfolyamatban fut
  - Külön folyamatban fut
- Erős mobilitásnál
  - Áthelyezett folyamat
  - Klónozott folyamat

Kódáthelyezési modellek (2)



A kódáthelyezés lehetőségei.

Az áthelyezés és a helyi erőforrások (1)

- Folyamat-erőforrás kötés
  - azonosítókapcsolt – kizárólag az adott erőforrás kell
  - tartalomkapcsolt – az adott erőforrás tartalma kell
  - típuskapcsolt – adott típusú erőforrás kell
- Erőforrás-gép kötés
  - csatolatlan – könnyen áthelyezhető
  - rögzített – áthelyezhető, de költséges
  - mozdíthatatlan – nem helyezhető át

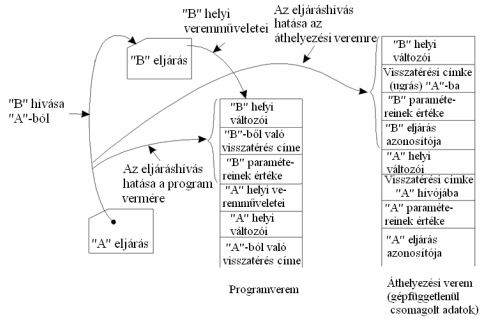
Folyamat-erőforrás kötés \ Erőforrás-gép kötés	Csatolatlan	Rögzített	Mozdíthatatlan
Azonosítókapcsolt	erőforrás áthelyezése (vagy globális hivatkozás létesítése)	globális hivatkozás létesítése (vagy erőforrás áthelyezése)	globális hivatkozás létesítése
Tartalomkapcsolt	tartalom másolása (vagy erőforrás áthelyezése vagy globális hivatkozás létesítése)	globális hivatkozás létesítése (vagy tartalom másolása)	globális hivatkozás létesítése
Típuskapcsolt	újrakapcsolódás helyileg elérhetőhöz (vagy erőforrás áthelyezése vagy tartalom másolása)	újrakapcsolódás helyileg elérhetőhöz (vagy globális hivatkozás létesítése vagy tartalom másolása)	újrakapcsolódás helyileg elérhetőhöz (vagy globális hivatkozás létesítése)

Másik gépre való kódáthelyezés során a helyi erőforrásokra való hivatkozásokkal kapcsolatos műveletek.

Kód áthelyezése heterogén rendszerben

- Szükséges az összes platform támogatottsága – újrafordítás
- Erős mobilitásnál – végrehajtási szegmens áthelyezhetősége
- Végrehajtási szegmens tartalma:
  - folyamat által elérhető védett adatok
  - programszámláló értéke
  - folyamat vermének tartalma
- Lehetséges megoldás:
  - kódáthelyezés csak meghatározott pontokon
  - programvermem másolatának (áthelyezési verem) tárolása gépfüggetlen módon
  - kódáthelyezéskor kód és áthelyezési verem átküldése

Kód áthelyezése heterogén rendszerben



A végrehajtási szegmens áthelyezhetőségének támogatásához szükséges áthelyezési verem kezelésének elvi vázlata heterogén környezetben.

Programozási nyelv által támogatott mobilitás – JoCaml

Szerver:

```
let def f x = reply x*x in
   Ns.register "square" f vartype;;

let loc here do {};;

Ns.register "here" here vartype;
Join.server ();;
```

Kliens:

```
let loc mobile do {
  let here = Ns.lookup "here" vartype in
    go here;

  let sqr = Ns.lookup "square" vartype in
    let result = sqr 5 in
      print_string ("square of 5 = "^string_of_int result^"\n");
      flush stdout;
}
```

Elnevezési rendszerek

1. rész

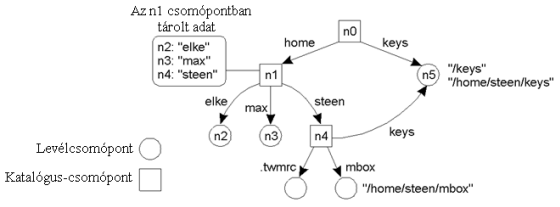
Nevek, azonosítók és címek

- Nevek – erőforrások megosztása, entitások egyértelmű azonosítása, helyek megjelölése, stb.
- Nevek feloldása – névszolgáltató rendszer
- Kapcsolódási pont
- Helyfüggetlen név
- Azonosító
- Emberközeli nevek

Névterek (1)

- Elnevezési gráfok
  - csomópontok
  - gyökér
  - globális név
  - helyi név

Névterek (2)

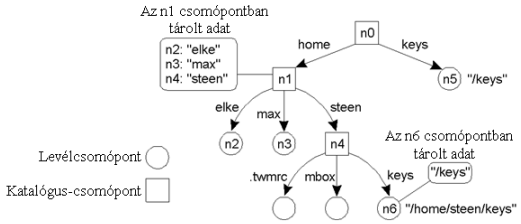


Általános elnevezési gráf egyetlen gyökércsomóponttal.

Névfeloldás

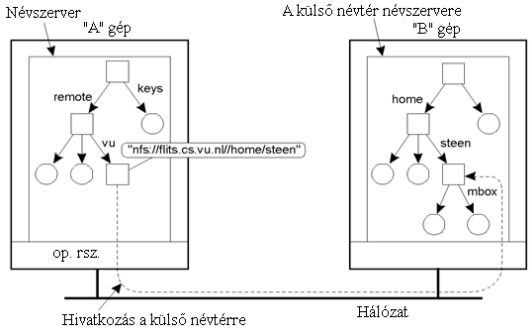
- Név visszakeresése
- Csomópontról csomópontra
- Kiválasztási mechanizmus

Csatolás és felszerelés (1)



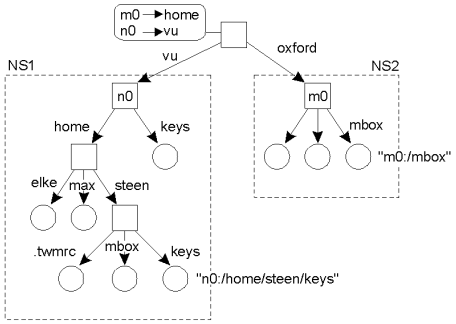
Az elnevezési gráfban lévő szimbolikus hivatkozás elvének magyarázata.

Csatolás és felszerelés (2)



Távoli névtér felszerelése adott hozzáférési protokoll segítségével.

Csatolás és felszerelés (3)



A DEC globális névszolgáltatásának felépítése.