Elosztott rendszerek minta ZH

**1. Hasonlítsa össze az elosztott es központosított rendszereket! (10p)**

|  |  |
| --- | --- |
| **Elosztott** | **Központosított** |
| Jól skálázható | egyszerűbb és biztonságosabb adminisztráció |
| Nagy rendelkezésre állású | képzett gárda |
| Lokálisan önálló | Megbízhatóság és elérhetőség |
| Terhelés elosztás |  |
| Nagyobb teljesítmény |  |
| Jobb ár / érték arány |  |
| Rugalmasság |  |

**2. Soroljon fel 10 middleware szolgáltatást es fejtsen ki ötöt részletesen! (15p)**

Távoli eljáráshívás

Szálkezelés

Tranzakciókezelés

Névfeloldás

Biztonság

Objektumok életciklusának kezelése

Terhelés elosztás

Aszinkron üzenetkezelés

Perzisztencia

Hibakezelés

**3. Mi a különbség az implicit es explicit middleware között? (10p)**

Implicit:

* Szétválik a fejlesztés és az adminisztráció
* Attribútumok( COM+)
* XML Descriptor ( EJB)
* Külön leíró fájl tartalmazza, milyen middleware szolgáltatásokat veszünk igénybe
* A Kérésmegszakító a leíró fájl alapján generálódik
* A forráskód valóban csak üzleti logikát tartalmaz
* A leíró fájlt módosíthatja a vevő, a forráskódot nem kell kiadnunk
* Egy lehetséges megvalósítás az interfész és implementáció szétválasztása

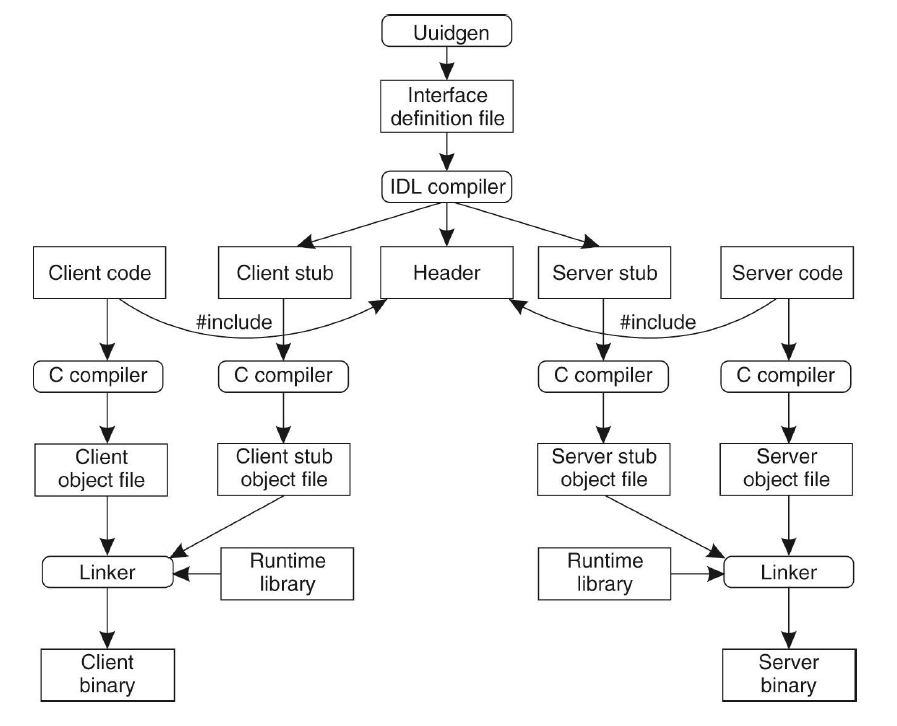
Explicit:

* Alapvetően API hívások révén
* Tipikusan a CORBA használja
* Felduzzad a forráskód
* Nem rugalmas a middleware (ha eladjuk a komponenst, ki kell adni a forráskódot, ha a vevő pl. más tranzakciókezelést akar)

**4. Soroljon fel 5 különbséget a COM es CORBA között! (10p)**

* COM csak Windows-on, CORBA platform független
* COM-ban nincs tranzakció kezelés, CORBA-ban van
* CORBA-ban kifinomultabb biztonsági modell mint COM-ban
* COM implicit middleware, CORBA explicit middleware
* COM root interfész: IUNKOWN, CORBA root: CORBAObject
* COM-ban egyszeres öröklés, másikban többszörös
* COM-ban az attribútumok nem támogatottak, CORBA-ban get;set;-re fordul

**5. Vázolja fel az RPC alapú fejlesztési folyamatot! (15p)**

****

**6. Mi az absztrakt szintaxis és mi a konkrét szintaxis a vizuális nyelveknél? (4p)**

Konkrét szintaxis: a szakterületi függő megjelenítés definíciója

Absztrakt szintaxis: a nyelv absztrakt jelentésének definíciója (amit a metamodellel adunk meg általában)

**7. Mit nevezünk external ill. mit nevezünk internal szöveges szakterületi nyelvnek? (4p)**

External: Külön specifikációs nyelvet használunk pl PictureProcessor nyelv

Internal: Egy programozási nyelvet használunk erre a célra, de nem az összes szimbólumát. pl script nyelvek

**8. Értékelje az alábbi állítást: A OCL egy gyengén típusos, imperatív nyelv, ami a MOF metamodellekben található és az elemek közti kapcsolatok megkötéseinek szabályait hivatott leírni. (7p)**

* Nem gyengén hanem erősen típusos
* Nem imperatív, hanem deklaratív
* Nem csak elemek közötti kapcsolatokat, hanem metamodell elemek belső működésére/állapotára is tehetők megkötések
* MOF metamodelleken használható, de hogy bennük található nem tudom mennyire helytálló

**9. Mik a denotációs szemantika alapelvei, módszerei? Szemléltesse őket röviden (!) a hexadecimális számok körében elvégzett négy alapművelet szemantikai leírásával! (10p)**

* A művelet hatására koncentrál
* Három részből áll:
  + Nyelv (a cél szakterület)
  + Szemantikai algebra (számítási modell leírására)
  + Leképző függvény

**10. Definiálja a következő nyelvet EBNF segítségével: bábuk vezérlése a sakktáblán. A bábuknak a színét, típusát és egyedi azonosítóját megadva meghatározhatjuk, hogy hova lépjen. Legyen lehetőség elágazások megadására az ellenfél bábuira való feltételek megadásával (pl. ha egy gyalog áll a B3-on, akkor lépjünk a C3-ra, különben a D4-re)! Csak érvényes mezőkód fogadható el, de az ütést nem kell kezelni a nyelvtanban. Opcionálisan lehessen megadni egy új bábu típust is a lépésnél, hogy modellezhessük, ha a gyalog beér az utolsó mezőre és átváltjuk egy másik bábura. Ne csak egy lépést lehessen leírni egy scripttel, hanem egy egész játszmát! (15p)**

**//Alap struktúrák definiálása**

**Számjegy = 0|1|2|3|4|5|6|7|8|9;**

**Szám = Számjegy – “0”, {Számjegy};**

**//Játék specifikus struktúrák**

**Azonosító = „ID\_”, Szám;**

**Mező = A|B|C|D|E|F|G|H , 1|2|3|4|5|6|7|8         ;**

**Típus = “gyalog”|”bástya”|”huszár”|”futó”|”király”|”vezér” ;**

**Szín = „világos”|”sötét”;**

**Bábu = (Azonosító,  ’’,’’ , Szín, ’’,’’ ,Típus)**

**Játszma = {Parancs};**

**Parancs = (HA | LÉP);**

**HA = “HA”, Feltétel, “AKKOR:”, Parancs, “KÜLÖNBEN:”, Parancs;**

**Feltétel = “BÁBU\_ÁLL”, (Mező, ’’,’’ , Típus);**

**LÉP = Bábu, „LÉPJEN”, Mező, [Típus]**

**//Példa játék, szicíliai védelem**

**(ID\_5,  világos, gyalog) LÉPJEN E4**

**(ID\_12, sötét, gyalog) LÉPJEN C5**

**//Feladat kiírás feltétel vizsgálata**

**HA BÁBU\_ÁLL (B3, Gyalog) AKKOR:**

**(ID\_7, világos, vezér) LÉPJEN C3**

**KÜLÖNBEN:**

**(ID\_7, világos, vezér) LÉPJEN D4**

**//Átalakulás példa**

**(ID\_11, sötét, gyalog) LÉPJEN A1, Vezér**