

“Magad, uram, ha szolgád nincs”

Elektronikus tanársegéd (ETS)



Web-alapon

<<http://www.inf.bme.hu/~dp>>

Benkő Tamás, BME SZIT & IQSoft Rt.

<benko@iqsoft.hu>

Hanák Dávid, BME VIK

<dhanak@inf.bme.hu>

Hanák Péter dr., BME IIT & OM KFHÁT

<hanak@inf.bme.hu>

Szeredi Péter dr., BME SZIT & IQSoft Rt.

<szeredi@iqsoft.hu>

Tartalom

1. Bevezetés
2. Az ETS rendszer áttekintése
3. Az ETS rendszer megvalósítása
4. Továbbfejlesztési tervek, lehetőségek

1. Bevezetés

1. Bevezetés

1.1. Motiváció, célok

1.2. Környezet: **Deklaratív programozás,**

BME Villamosmérnöki és Informatikai Kar

1.3. Hasonló, illetve rokon rendszerek

2. Az ETS rendszer áttekintése

3. Az ETS rendszer megvalósítása

4. Továbbfejlesztési lehetőségek és tervek

1.1. Motiváció, célok

- Deklaratív programozás 1995 óta az egész műszaki informatika évfolyamnak a BME-n
- A tárgyhoz nincs gyakorlati, sem laborfoglalkozás
- Programozni tanulni **csak gyakorlással** lehet
- A hallgatók **önálló munkájára** építünk, amit serkenteni és segíteni kell:
 - otthon használható programozási környezettel,
 - önálló feladatok kiadásával,
 - **ellenőrzéssel, beszámoltatással, értékeléssel,**
 - a tanári segítséget pótló gyakorlási lehetőséggel

1.1. Motiváció, célok (folyt.)

Az elektronikus ellenőrzés pozitívumai

- “vasszigor”: **következetesség a specifikáció betartatásában,**
- pontozáson alapuló **objektív értékelés,**
- algoritmusok **végrehajtási idejének** mérése,
- az értékelés felgyorsulása,
- az oktatói terhelés csökkenése.

Megoldás: **elektronikus tanársegéd** (ETS)

a hallgatók és az oktatók munkájának segítésére

1.1. Motiváció, célok (folyt.)

További tényezők:

- Meredeken emelkedő hallgatói létszám
~100/1995 ... 360/2001
- Asszisztencia hiánya
Vannak: felsőbb évfolyamos demonstrátorok,
egy-két doktorandusz hallgató,
diplomatervezők

Megoldás: **elektronikus tanársegéd** (ETS)
a hallgatók és az oktatók munkájának segítésére

1.1. Motiváció, célok (folyt.)

Az elektronikus ellenőrzés kényszerű negatívumai:

- jóval kevesebb a személyes interakció a hallgatókkal
 - a személyes hatás gyengül
(ez elsősorban a létszámnövekedés következménye),
- a hallgatók programozási stílusát kevésbé ellenőrizzük, alig kritizáljuk → kevésbé fejlesztjük,
- az elektronikus beadás másolásra (csalásra) csábítja a hallgatók egy részét.

1.2. Környezet: Deklaratív programozás

- Imperatív vs. deklaratív programozás
 - *imperatív*: felszólító módú, utasításokból áll; változófogalma: változtatható tartalmú memóriahely,
 - *deklaratív*: kijelentő módú, egyenletekből, állításokból áll; változófogalma: ismeretlen, de (előbb--utóbb) rögzített értékű mennyiség
- A deklaratív programozás főbb tanulságai
 - jelszó: MIT és nem HOGYAN; a megoldás módja helyett a megoldandó feladatot kell leírni,
 - a ciklusok írása helyett absztrakcióra, már megoldott feladatra való visszavezetésre (*rekurzióra*) nevel

1.2. Környezet: Deklaratív programozás

- Tárgykövetelmények
 - **programozási nagyfeladat** - SML, Prolog, közös dokumentációval
 - **nagyzárthelyi** (pótzárthelyi)
 - vizsga (korábban írásbeli, 2000-től szóbeli)
 - **kisfeladatok** (2000-től: 3-3, nem kötelező)
 - **gyakorló feladatok** Web-felületen (kísérleti 2001-ben)
- **Létraverseny** a leggyorsabban futó programoknak
- A hallgatók túlterhelése
 - fennálló veszély, örök dilemma
 - részmegoldás: a terhelés időbeli elosztása

1.3.Hasonló, illetve rokon rendszerek

- Hasonló rendszerek?
 - nem találkoztunk igazán hasonlóval
- Távolabbi rokonok
 - Intelligent Tutoring System (ITS),
 - Web Based Training (WBT),
 - Computer Aided Learning & Training (CAL, CAT),
 - GIML: Gentle Introduction to ML
 - hiperszöveges SML-tankönyv gyakorló kérdésekkel,
 - programozási versenyek (ACM, IOI, ...)

2. Az ETS rendszer áttekintése

1. Bevezetés

2. Az ETS rendszer áttekintése

2.1. Az ETS-sel megoldandó feladatok

2.2. Az ETS rendszer architektúrája

3. Az ETS rendszer megvalósítása

4. Továbbfejlesztési lehetőségek és tervek

2.1. Az ETS-sel megoldandó feladatok

Oktatás:

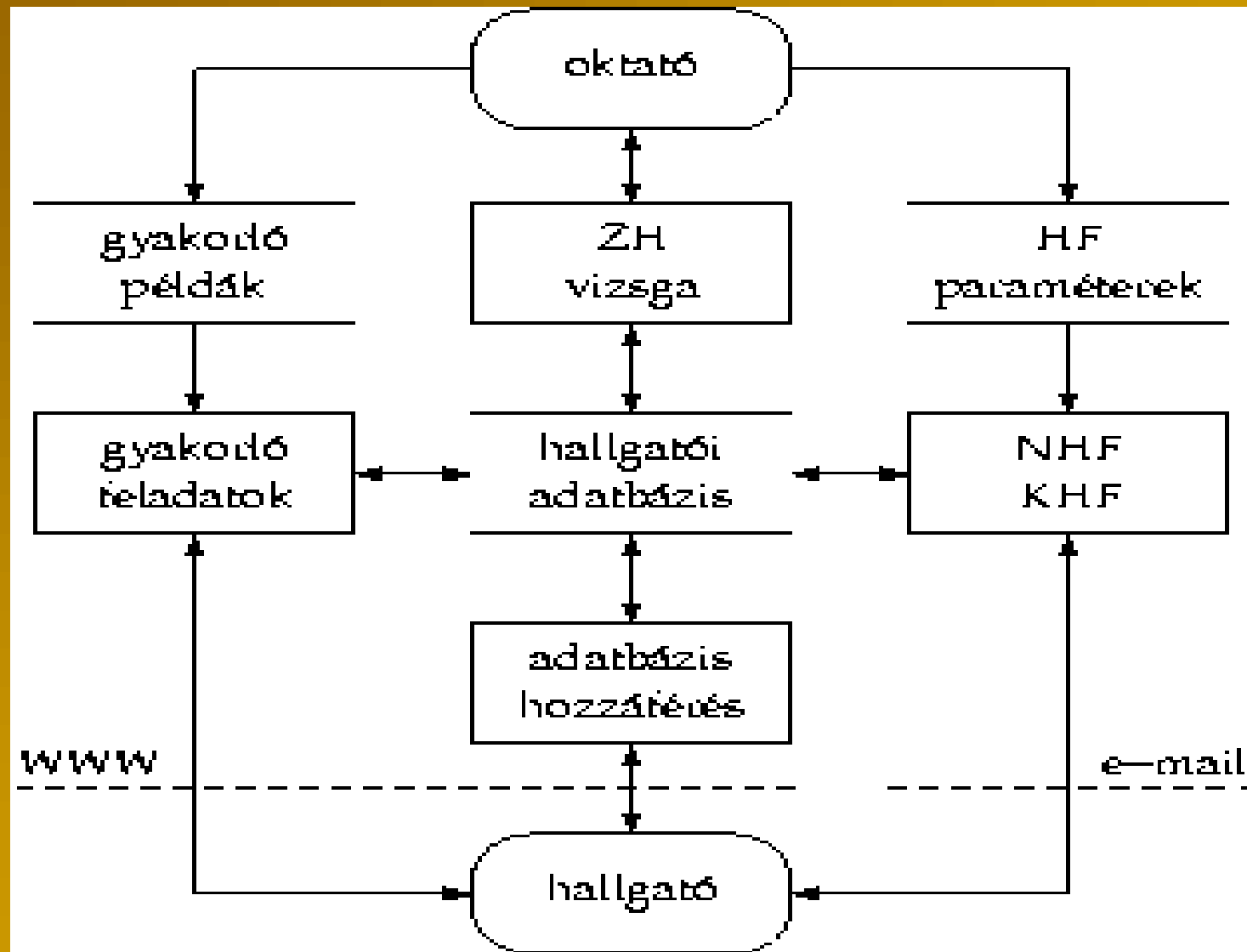
- programozási (nagy, kis) **házi feladatok**
fogadása elektronikus levélben, tesztelése, pontozása,
hallgatóknak az eredmények elküldése,
- **gyakorlórendszer** a tárgy tananyagának
elsajátításához (vö. WBT),
felkészülés segítése zéhára, vizsgára,
- a **programmásolás** felderítése,
- a zárthelyik névre szóló előállítása és értékelésének
támogatása

2.1. A megoldandó feladatok (folyt.)

Adminisztráció:

- a tárgyadminisztráció segítése
részpontok, pontok tárolása, összegzése, hallgatók tájékoztatása, vizsgára pontozólapok előállítása, statisztikakészítés
- a levelezési lista működtetése
- a jegyzet-, zéhá- stb. jelentkezések adminisztrálása
- a tárgyhonlap karbantartásának támogatása
dinamikus tartalom

2.2. Az ETS rendszer architektúrája



3. Az ETS rendszer megvalósítása

1. Bevezetés

2. Az ETS rendszer áttekintése

3. Az ETS rendszer megvalósítása

3.1. A házi feladatok feldolgozása

3.1.a. Első változat: reaktív programok

3.1.b. Második változat: nem-reaktív programok

3.1.c. Kis és nagy házi feladatok feldolgozásának
egységesítése

3.2. Gyakoroltató rendszer

3.3. Másolás elleni védelem

3.4. Egyéb komponensek

4. Továbbfejlesztési lehetőségek és tervek

3.1. A házi feladatok feldolgozása

- Általános, nehezen megkerülhető nehézségek:
 - **csúcsterhelés** a beadási határidő előtti órákban,
 - **lassú** vagy **végtesen ciklusba eső** programok
 - meghatározott idő (30-180 s CPU-idő) után kilőjük,
 - 5-15 CPU-percig tarthat egy-egy program tesztelése
 - 10 tesztesetből álló tesztkészlet használata mellett
- Az automatikus válaszadás előnye:
 - a határidő leteltéig a **hibák kijavíthatók**, a programok többször beadhatók: csak a legutolsó verzió számít

3.1.a. Az ETS rendszer első változata

Első változat: **reaktív programok** (1995, 1996)

- **Párbeszéd** szöveges felületen:
két **processz** (az aktív és a passzív „játékos”) beszélget egymással, rögzített protokoll szerint
- Példák: **Mastermind**, Minesweeper
- Fogadás elektronikus levélben, **tesztelés azonnal, utána értesítés** elektronikus levélben
- Párbeszédet közvetítő **processz** („piperun”) a standard input és output között
- Futtatási **eredmények szöveges állományokba**, feldolgozásuk kézzel és segédprogramokkal

3.1.a. A reaktív programok problémái

- I/O-ra várakozás **protokoll be nem tartása** miatt
 - nem fogy a CPU-idő, nincs időlimit-túllépés, de van
 - felesleges tárfoglalás, fokozódó túlterhelés ← swapping
- **Hibák** egy része **fordításkor felderítetlen** marad
- A reaktív párbeszéd otthon **nehezebben tesztelhető**
- A teszteléshez használt **keretprogram erősen függ a feladattól**
- A protokoll (specifikáció) következetes betartatása SE-szempontról tanulságos, de
 - munkaigényes, hibákra érzékeny,
 - kevés a köze a deklaratív programozáshoz

3.1.a. Az első ETS-változat problémái

- Minden levélben beérkező program **azonnal** két **újabb processzt** indított el – ez túlterhelhette a számítógépet
- Elektronikus **válaszlevelet** csak a tesztelési **eredmények elkészülte után** küldtünk:
 - a türelmetlenek újból és újból elküldték a feladataikat,
 - feleslegesen nőtt a terhelés.
- A **generált állományok** – egyes programok “debug” kimeneteitől – **hatalmas méretűre** duzzadhattak:
 - ilyenkor a hallgató esetleg nem kapta meg a válaszlevelet,
 - ezért újra elküldte a feladatot, sít.

3.1.a. Az első ETS-változat problémái (f.)

- A ritkán (továbbfejlesztéskor) és a gyakran (félévente szükségszerűen) változó **részek nem voltak szétválasztva**
- A különböző operációs rendszerek (Solaris, Linux) alatti **bináris kódváltozatok ne voltak szétválasztva** (vö. hordozhatóság: otthoni tesztelés, vizsgán ellenőrzés)

Az első verziónál tapasztalt nehézségeket azóta nagyrészt megoldottuk vagy túlhaladottá váltak.

3.1.b. Az ETS-rendszer második változata

1997 óta

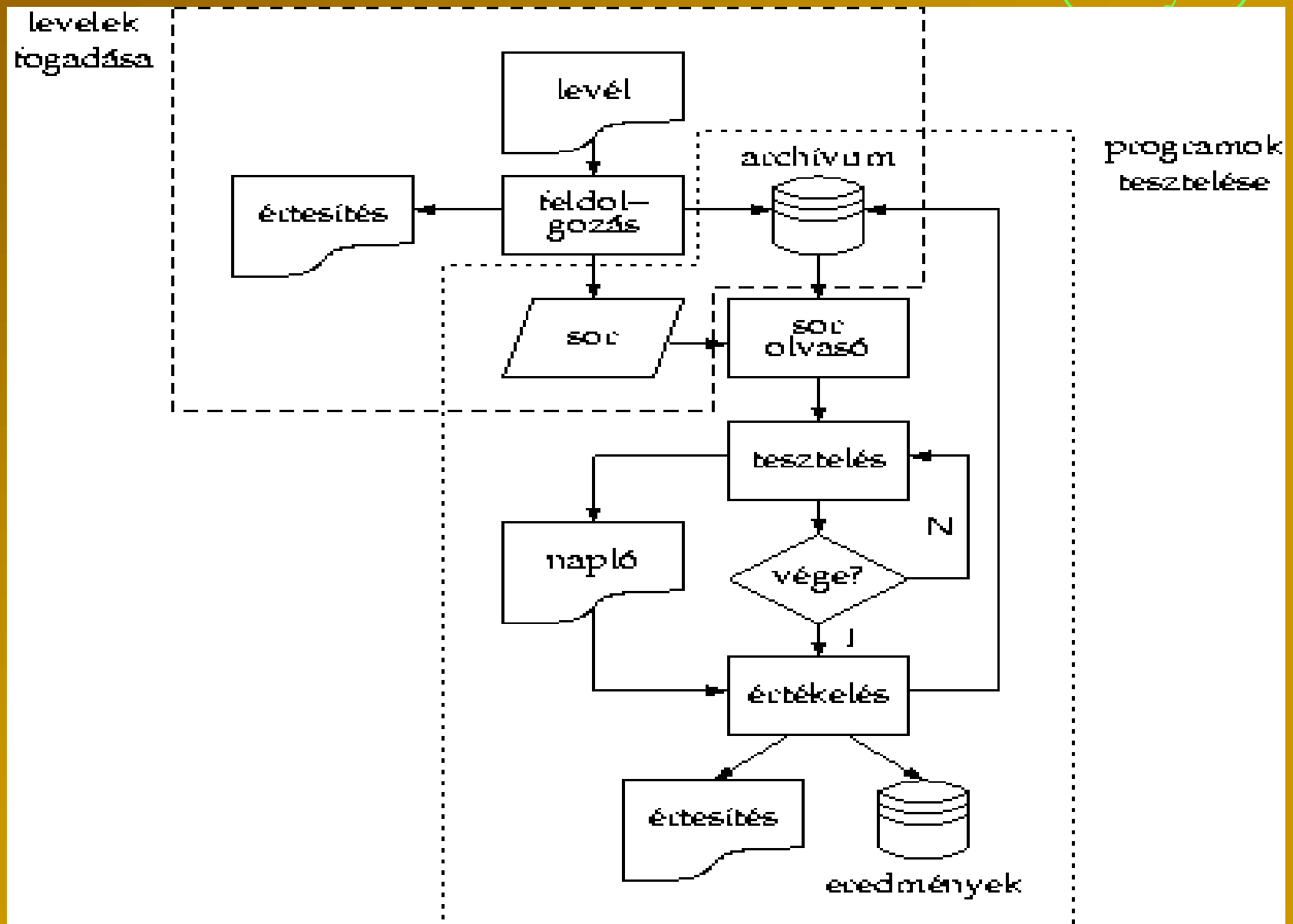
- nem reaktív programot, hanem
- adott specifikációt kielégítő *függvényt* (SML), ill. *eljárást* (Prolog) kell írni.
- Példák: téglalap-kirakós, **kitalálós** (brain), **dominó-parketta**, sátoertábor
- Fogadás elektronikus levélben, **nyugtázással**, utána **tesztelés**, majd **értésítés** elektronikus levélben

3.1.b. A második ETS-változat

Javítottuk a megbízhatóságot és karbantarthatóságot:

- **Egyszerre csak egy** programot tesztelünk, a többi várakozó sorban vár a sorára
- A levélben beérkező programokat “kicsomagolás” után **a várakozó sor végére** tesszük
- Minden beérkező levelet **azonnal nyugtázzunk**, és **közöljük a sor hosszát** (továbbfejlesztés: időbecslés, szükség esetén finomított ismétléssel)
- A tesztelés **után az eredményről újabb levélben** értesítjük a feladót
- Az állandónak tekinthető és a félévente változó **részeket különválasztottuk**

3.1.b. A második ETS-változat (folyt.)



3.1.c. Kis házi feladatok feldolgozása

- 2000-től kezdve kis házi feladatokat is adunk ki
- Fogadásuk, tesztelésük nagyon hasonló a nagyfeladatokéhoz
- Rövid távú terv: a kis és a nagy házi feladatok feldolgozásának egységesítése

3.2. Gyakoroltató rendszer

- Web-felületű Prolog- és SML-gyakorló rendszer
- a zéhán és a vizsgán előforduló típusfeladatok gyakorlására
- Prolog: (1) Prolog-kifejezés kanonikus alakja, (2) egyesíthetőség vizsgálata, (3) írjon olyan eljárást, amely ...
- SML: (1) típusegyenletek, (2) mi a függvényalkalmazás eredménye, (3) írjon olyan függvényt, amely ...

3.3. Másolás elleni védelem

- Az elektronikus beadás és automatikus tesztelés **másolásra csábítja** a hallgatókat
- A programszöveget csak felületesen tudjuk átnézni, szűrőpróbaszerűen ellenőrizni
- Az **időadatok összevetése** sok esetben jelzi a másolást
- A **programok hívási fájának összehasonlítása** nagyobb biztonsággal, analitikusan is felismerhetővé teszi a másolást
- A **talált egyezéseket felül kell vizsgálni**, ui. lehetnek “vakriasztások” bizonyos – egyes másolási trükköket leleplező – redukciós lépések miatt

3.3. Másolás elleni védelem (folyt.)

Forráskódú (Prolog-) programok hasonlóságának vizsgálatára 2000-ben program készült

- Lukácsy Gergely műszaki informatikus hallgató,
- TDK-dolgozat: 2000-ben rektori különdíj, 2001-ben országos I. díj
- konzulens Szeredi Péter és Benkő Tamás,

A hasonlóságot vizsgáló programhoz SML-programok hívási fáját felépítő modul készül

3.4. Egyéb komponensek

- Adatbázis
 - házi feladatok, zárthelyik eredményei szöveges állományokban
 - létraverseny eredményei szöveges állományokban
- Segédprogramok
 - Web-lapon megnézhető eredmények előállítására
 - jegyzetrendelésre, zárthelyire való jelentkezésre
 - vizsgalapok előállítására
 - vizsgaadminisztráció segítésére

4. Továbbfejlesztési tervek, lehetőségek

1. Bevezetés

2. Az ETS rendszer áttekintése

3. Az ETS rendszer megvalósítása

4. Továbbfejlesztési tervek, lehetőségek

4. Továbbfejlesztési tervek

- Házi és gyakorló feladatok, zárthelyik, vizsga stb. **eredményeit** szöveges állomány helyett **adatbázisba** tesszük
- A gyakorlásra szolgáló **feladattípusok körét bővítjük**, a gyakorlás tényét regisztráljuk
- **Becslést** adunk **a tesztelés várható időpontjára**, és ha szükséges, később finomítjuk
- **A programrendszer elemeit** (most nagyrészt bash-szkriptek) újraírással **karbantarthatóbbá** tesszük

4. Továbbfejlesztési lehetőségek

Hosszabb távon:

- A zárthelyi-feladatsorok névre szóló előállítását adatbázisból elővett típusfeladatok véletlenszerű kombinálásával,
- A zárthelyik javításának számítógépes támogatása
- A deklaratív programozáshoz kötődő és nem kötődő részek szétválasztása

4. Továbbfejlesztési lehetőségek (folyt.)

- Házi feladatok elosztott ellenőrzése több gépből álló rendszeren
- Lehetséges stratégiák
 - véletlenszerű,
 - első teszteset ideje szerint osztályozva,
 - az előző verzió összideje szerint osztályozva
 - egyenletes terhelésre törekedve,
 - gyors(abb) programokat jutalmazva (előre hozva)
 - ez egyetlen gépen is működik

4. Továbbfejlesztési lehetőségek (folyt.)

- A teljes rendszer újratervezésével diplomaterv foglalkozik
- Az egyes részfeladatokat önálló laboratórium, demonstrátori és TDK-tevékenység keretében hallgatók bevonásával oldjuk meg

Közreműködők

Köszönetet mondunk

- Péter Lászlónak,
- Szeredi Miklósnak,
- Gefferth Andrásnak,
- Rozmán Tamásnak

a BME VIK egykori hallgatóinak egyes részek
kidolgozásáért