

Egy komplex példa

- **Puttycd e glite-tutor.ct.infn.it vagy glite-tutor2.ct.infn.it**
- **Felhasználói név: budapest01....15**
- **Jelszó: GridBUD01....15**
- **Jelszó a tanúsítványhoz: BUDAPEST**

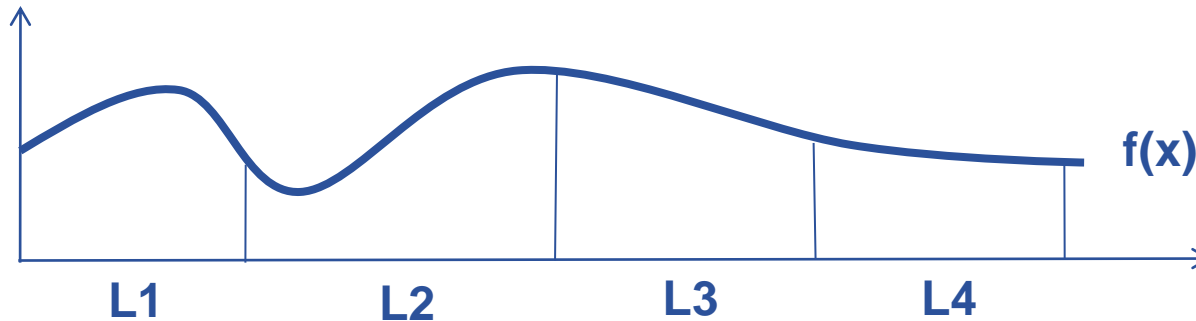
- **Feladatok letöltése:**
 - `wget http://bagira.iit.bme.hu/~nemeth/example.tgz`
 - `tar xfz example.tgz`
 - ...

- `glite-wms-job-submit -a brumm.jdl`
- `glite-wms-job-status JOBID`
- `glite-wms-job-output JOBID`
- `glite-voms-proxy-init --voms gilda`
- `glite-wms-job-list-match -a brumm.jdl`
- `glite-voms-proxy-info --all`
- `mc, vim, watch, wget`

1. Feladat – Hello World!

- **Megtalálható a example1 könyvtárban**
- **A parancssori toolok gyakorlása**

- Szöveg: Integráljuk az $f(x)$ függvényt úgy hogy az értelmezési tartományt $L1, L2, \dots, LN$ részekre osztjuk, majd ezen részintegrálok eredményét összegezzük



- **Feladat szinten: Legyen $N=3$**

```
#include <stdio.h>
```

```
double f(double x){
```

```
    return x * x + 3 * x + 5;
```

```
}
```

```
int main(int argc, char ** argv){
```

```
    //args start length resolution start
```

```
    int buffer;
```

```
    sscanf(argv[1], "%d", &buffer);
```

```
    double start = 0.0 + buffer;
```

```
    sscanf(argv[2], "%d", &buffer);
```

```
    double end = start + buffer;
```

```
    double resolution = 1000;
```

```
    sscanf(argv[3], "%lf", &resolution);
```

```
    double f_start = f(start);
```

```
    double sum = 0.0;
```

```
    while(start < end){
```

```
        sum += f(start) * resolution;
```

```
        start += resolution;
```

```
    }
```

```
    printf("%lf", sum);
```

```
    return 0;
```

```
}
```

```
#!/bin/bash
```

```
gcc main.c -o main &> compile.log
```

```
if [ $? -ne 0 ]
```

```
then
```

```
    echo "Hiba a fordítás során"
```

```
    exit
```

```
fi
```

```
./main $1 $2 $3
```

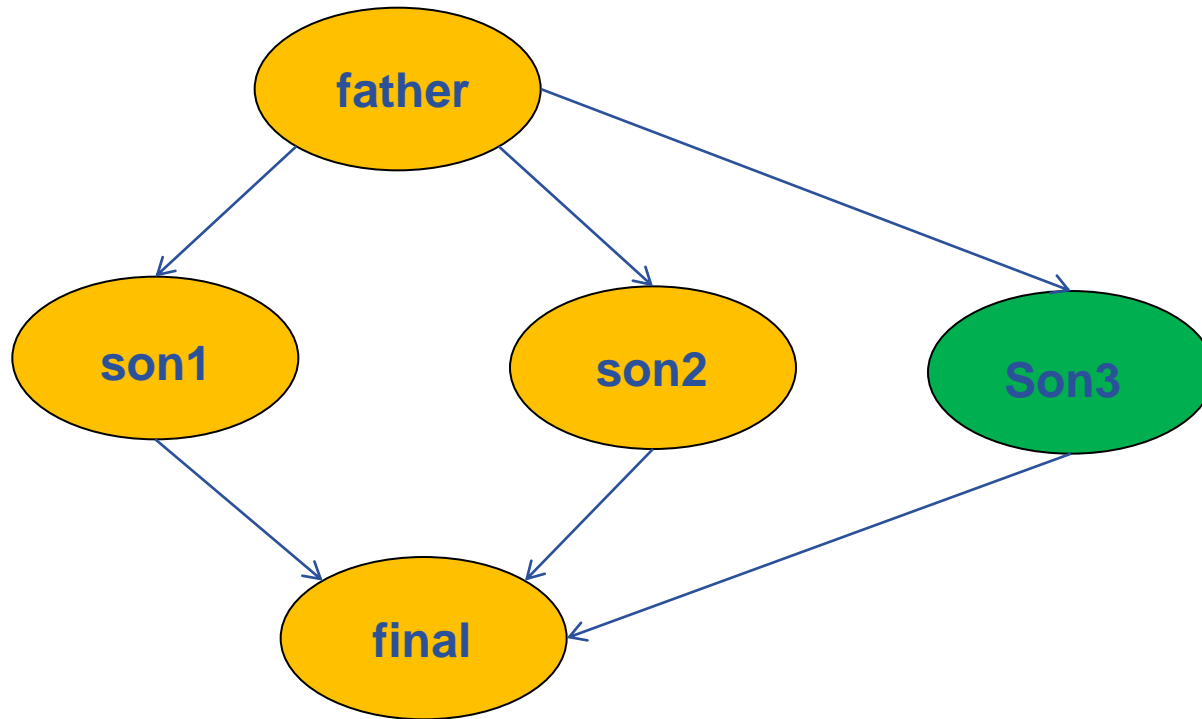
- Feladateleíró: (JDL)

[

```
JobType = "Parametric";  
Executable = "wrapper.sh";  
Arguments = "_PARAM_ 2 0.01";  
InputSandbox = {"wrapper.sh", "main.c"};  
Parameters = 6;  
ParameterStep = 2;  
ParameterStart = 0;  
StdOutput = "integ.out._PARAM_";  
OutputSandbox = {"integ.out._PARAM_"};  
ShallowRetryCount = 1;
```

]

- Vegyen fel egy új csomópontot
 - Ehhez nem feltétlenül csak a JDL-t kell módosítani



- Adott egy prím szám (multiplied.prims)
- P1. Example (0) :
- 78484762303185332742698040745323489384931382031
386791984783233501593772118861
- Bontsuk fel két prímszám szorzatára, **ha tudjuk, hogy ezek csak a megadott fájlokban szereplő számok lehetnek**
- Mindenkinek a csak a kapott sorszámú prímet kell felbontani

- **Elérés:**

1. Scripből

- `export LCG_GFAL_INFOSYS=glite-rb.ct.infn.it:2170`
- `export LCG_CATALOG_TYPE=lfc`
- `export LFC_HOST=lfc-gilda.ct.infn.it`
- + `lfc-s` parancsok (lásd előző fájl.ppt)
- 5 fájl kell hozzá amik valhol a `/grid/gilda/tutorials/budapest` könyvtárban vannak

2. JDL-en keresztül (de most nem ezt használjuk)

- **Keresse meg a budapest kezdetű könyvtárat**
 - A nevezési konvenciónak megfelelően a tutorials könyvtárban
- **Van benne néhány fájl azok tartalmazzák a primmeket**

- `./solver.sh`
78484762303185332742698040745323489384931382
031386791984783233501593772118861
- PRIM1 :
299995112848418750555290768312301913947
- PRIM2 :
261620136268159941252512381313325676663