

# Tema 3 ASC - Bazele arhitecturii Cell

## Prison Break

**Deadline 18.05.2008 ora 23:55**

### Prezentare

Intr-un labirint pregenerat (sau generat aleator pentru un mic bonus), este plasata o echipa de 8 prizonieri aleatori in labirint (nu neaparat ca un grup ci ca indivizi). Exista o singura iesire, si exista drum de la fiecare prizonier la iesire. Fiecare prizonier cauta iesirea din labirint, putandu-se deplasa in orice directie in care nu se afla un perete. In unele locuri ei trebuie sa depaseasca niste usi cu o combinatie (de fiecare data cand trec prin usa). O combinatie este reprezentata de un sistem de  $N$  ecuatii cu  $N$  necunoscute, nebanala.  $N$  va fi suficient de mare ca o prelucrare folosind SIMD sa fie indicata.

Odata ce unul gaseste iesirea, toti prizonierii trebuiesc sa se deplaseze pana la iesire si acolo sa sparga codul de la usa de iesire, in echipa de data aceasta. Codul de data aceasta se constituie din sortarea unui vector  $f$  mare, impartindu-se problema in subprobleme, fiecare sortand un vector si apoi interclasandu-se, vectorul va fi de dimensiune  $M$ .

### Cerinte:

Generarea (sau citirea/pregenerarea) labirintului, a vectorului de sortat si a sistemelor de ecuatii se face de catre PPU. Comunicarea intre SPU si PPU-uri se va face prin MailBoxes, folosind event-uri. Comunicarea sistemelor de ecuatii de la obstacole si a vectorului de la iesire catre SPU se va face folosind DMA. La fel si pentru comunicatia inversa se va folosi DMA.

Pentru simplitate sugeram folosirea eliminarii gaussiene cu/fara pivotare pentru a rezolva sistemele. Solutii mai eficiente vor fi recompensate.

Programul va primi ca parametrii marimea labirintului (ce va fi patrat) si  $N$  si  $M$ .

Programul va crea la iesire 1) drumul prizonierului care a gasit iesirea 2) drumurile celorlalti prizonieri.

Readme-ul va contine explicatii de cum a fost rezolvat sistemul de ecuatii si de modul in care se face depasirea obstacolului final.

### COMPLETARI SI LAMURIRI

1) O dimensiune buna pentru  $N$  mi se pare 64

1) O dimensiune buna pentru  $M$  mi se pare  $4096 \cdot 8$

1) O dimensiune buna pentru L (latura labirintului) mi se pare 16 daca alegeti sa aveti labirint "garden maze" sau 8 daca alegeti labirint "walled chambers" sau 4 daca labirintul este in trei dimensiuni (oricare din implementari). Pentru alte implementari ale labirintului pe care eu nu mi le pot imagina in acest moment daca nu reusiti sa extrapolati din valorile de mai sus trimiteti un e-mail cu o descriere a ce vreti sa faceti si o sa va dau o dimensiune.

"garden maze" = labirint impartit in patrate, unde un patrat poate fi plin sau gol.

"walled chambers" = labirint impartit in camere, fiecare putand avea sau nu usi catre camerele adiacente.

Explicatie:

Scopul nu este sa testam sa vedem la ce valoare a lui N/M o sa spuna SPU-ul ca nu mai are spatiu in memoria locala si nici sa trebuiasca sa pierdeti minute bune cu fiecare rulare, ci ca N si M sa fie suficient de mari ca efortul suplimentar reprezentat de prelucrarea SIMD sa aiba sens. (de exemplu pentru  $n = 4$  si  $m = 32$  efortul nu s-ar justifica.

2) Programul are trei intrari : L latura labirintului patrat, N numarul de ecuatii si necunoscute, M marimea vectorului de la sfarsit.

Voi trebuie sa va generati singur in program atat labirintul si sistemele de ecuatii cat si vectorul de sortat (este OK daca faceti asta in PPU inainte de a plasa prizonierii).

Output-ul va trebui sa contina:

( in no particular order)

1) labirintul initial (incercati sa gasiti o forma de reprezentare sugestiva) pe care sa figurati pozitiile prizonierilor, a capcanelor si a iesirii.

2) pentru o ecuatie: ecuatia initiala si solutiile gasite

3) vectorul nesortat si dupa aceea vectorul sortat

4) drumul prizonierului care a gasit iesirea si dupa aceea si drumurile celorlalti.

Readme-ul va trebui sa contina explicatii despre cum ati facut fiecare din elementele de mai sus, si despre formatul output-ului.

### **ATENTIE:**

-10p: folosirea altui sistem in afara de event-uti pentru comunicatie (de ex. busy waiting)

-10p: nefolosirea instructiunilor specifice SIMD

-20p: neimplementarea obstacolului final (adica daca doar un prizonier gaseste iesirea)

## **BONUS-uri**

+10p: Labirint generat aleator (trebuie sa aiba cale de la fiecare prizonier la iesire si o singura iesire)

+10p: Folosirea DMA double buffering unde aceasta posibilitate exista

+10p: Folosirea unor algoritmi deosebit de eficienti in a depasi obstacolele (Sisteme(5p) + Vector final (5p))

+10p: mod usor inteligibil de a afisa drumul prizonierilor

## **Link-uri**

<http://www.ucdc.info/cd/doc/1258/5.pdf> - document despre ecuatii liniare pentru cei care nu isi mai aduc aminte de matematica din anii mai mici si din liceu 😊

[http://fmi.unibuc.ro/cniv/2006/disc/cniv/documente/pdf/sectiuneaB/6\\_16\\_popovici.pdf](http://fmi.unibuc.ro/cniv/2006/disc/cniv/documente/pdf/sectiuneaB/6_16_popovici.pdf) - un document despre ecuatii liniare (Partea matematica poate fi interesanta si utila).

<http://citeseerx.ist.psu.edu/showciting;jsessionid=A92C836560F6FA5F689EB7B9A3B2D216?cid=73205> -mai multe link-uri care contin metode eficiente si paralele de rezolvare e ecuatiilor liniare dense.

Ultimul link de pe acea pagina este cel mai util din punctul nostru de vedere:

<http://citeseerx.ist.psu.edu/viewdoc/summary;jsessionid=49A5B775BA2737A88A4D07E1C73426D6?doi=10.1.1.45.2825>

Modificari mai pot aparea numai in urma intrebarilor si observatiilor voastre.