

Eerste Deeltoets Concurrency

2 oktober 2012, 11.00–13.00, Educ-Gamma.

Motiveer je antwoorden *kort!* Zet je mobiel uit. Stel geen vragen over deze toets; als je een vraag niet duidelijk vindt, schrijf dan op hoe je de vraag interpreteert en beantwoord de vraag zoals je hem begrijpt.

Cijfer: Vragen 1 t/4 zijn elk 2pt en vragen 5 t/m 7 elk 4pt, samen 20, totaal delen door 2.

- Beantwoord met Amdahls Regel:** Een programma bestaat voor 80% uit parallesiseerbare code en wordt gedraaid op een quad-core.
 - Welke speedup wordt gehaald?
 - Om de executie verder te versnellen kunnen we kiezen tussen (1) het programma verbeteren zodat 90% parallesiseerbaar is; (2) meer cores bijschakelen. Hoeveel cores heb je bij optie (2) nodig om dezelfde speedup te halen als bij optie (1)?
- Safety of Liveness:** Zeg van elk van deze uitspraken of ze een *Safety* of een *Liveness* eigenschap beschrijven en waarom (in 1 zin).
 - De relschoppers worden opgespoord, berecht en gestraft!
 - Asielaanvragen worden binnen zes maanden afgehandeld.
 - Als een of meer threads aan de operatie beginnen, zal tenminste een de operatie in eindige tijd kunnen voltooien.
 - Als een thread t_2 de ticket-keuze begint nadat thread t_1 de ticket-keuze heeft voltooid, ontvangt t_2 een hoger ticket dan t_1 .
 - De scheduler voor threads t_1 , t_2 en t_3 is fair.
- Schoppen:** Klaverjassen wordt gespeeld met 32 kaarten, waarvan 8 van type Schoppen. Je begint met een “hand” van acht kaarten. Wat is de kans dat je begint met precies zes Schoppen-kaarten?
- Drie waarden gooien:** Je gooit herhaald met een dobbelsteen (zes-kantig) en gaat hiermee door totdat je *drie verschillende* uitkomsten hebt gezien. Wat is het verwachte aantal keren dat je moet gooien?
- LockTwo:** Hier staan de lock en unlock van de LockTwo klasse (voor thread i).

```
public void lock()                public void unlock()
{ victim = i;                    { }
  while (victim == i) {} } }
```

 - Aan welke drie eisen moet een lock implementatie voldoen?
 - Welke van deze eisen is/zijn voor LockTwo niet voldaan? Waarom?
 - Als je de opdracht `victim = i` verplaats naar de `unlock`, is dan het probleem opgelost?
- Assignments:** In een situatie waar x is 0 en v is true, worden deze twee threads opgestart:

```
Thread 1:      Thread 2:
  x = 1;        while (v) {};
  v = false;    y = x;
```

Is het zeker dat de waarde van y na afloop 1 is wanneer

- read/write atomicity geldt? Waarom?
- dit wordt uitgevoerd in Java of C#? Waarom?
- v als `volatile` wordt gedeclareerd?

7. **Multivalued register:** Je kunt een m -waardig *regular* register maken uit een array van m regular bits. De Writer schrijft waarde x door bit x op 1 te zetten en de lagere bits op 0, de reader zoekt vanaf positie 0 naar de eerste 1:

```
Write(x):                Read:
  r[x].write(1)          for(i=0; i<m; i++)
  for (i=x-1; i>=0; i--)  if (r[i].read == 1)
    r[i].write(0)         return i
```

- (a) Laat zien dat de Reader kan falen wanneer de Writer ook de hogere bits op 0 zet (met een loopje `for (i=x+1; i<m; i++) r[i].write(0)`).
- (b) Is het gebouwde register ook atomic? Leg uit!

¹<http://msdn.microsoft.com/en-us/library/aa645755%28v=vs.71%29.aspx>

²<http://stackoverflow.com/questions/6526432/volatile-with-release-acquire-semantics>