Notities 28/10/2016 TAGP - Examenopdracht

**Principe**

* Simulator: discrete event simulator
* 1 resource -> deze laat ge zichzelf simuleren

1 activiteit -> reserveert de resource

**Versie a:** deterministisch, simulator doet wat de resource vraagt

**Versie b:** stochastisch, de processen moeten met onzekerheden en veranderingen overweg kunnen

* Vooruitkijken: reservering en dergelijken + mogelijkheid om planning te herzien
* Initiatief: “een werknemer die initiatief neemt”, we mogen zelf kijken hoe we dit verder uitwerken  
  -> Web, google kalender (Cowboy, nog een introductie van L. Rutten krijgen)  
  -> n activiteiten en n resources, coördinatie systeem

Dit moet gepresenteerd worden voor het examen. Het vierde punt “initiatief” is voor extra punten, de rest is het minimum dat bereikt moet worden. 50% van het totaal examen.

**Schema Simulator**

{{tijdstip, make.ref} …………}  
make.ref = een unieke waarde van de taak; een seriële code.  
loop(starttijd A) -> Receive, events ontvangen die in tabel A (ETS tabel) moeten komen. Post Increment, “steek in de tabel dat ik binnen 3 seconden aankom”.   
Je rekent uit wanneer uw volgende event is en dan pas voer je die uit.  
 -> bereken tijd (T); after T get event from A.  
Gebruik steeds een nieuw proces om een event uit te voeren.

Loop kijkt wanneer de volgende te simuleren event is. Klanten kunnen events posten terwijl de loop bezig is.

**Versie a:** {Pid (processID), Message}; Message is een bevestiging of een event al verlopen is of niet.  
Pid!{Message,tijd}

**Versie b:** {r,f,message}. De klant van de simulator beslist wat er gebeurd wanneer er een event uitgevoerd wordt. M:F(a1,a2…) 🡪 spawn(M,F,Args)

**Versie c:** Anonieme functies, clone {AF} -> AF().

Zorg dat **versie a** werkt, ga dan verder met **versie b** en als er tijd over is, **versie c.**

**F(Pid,…) ->  
 Pid!......,  
 Receive  
 ……  
end**