

# Ontwerp en evaluatie van een process scheduler

Vincent Naessens - 16 februari 2021

---

## 1 Doelstellingen

In dit practicum worden een aantal scheduling strategieën geïmplementeerd in Java, zoals gezien in de theorielessen. De scheduler wordt getest met een XML-dataset van processen en hun parameters. De bekomen resultaten worden vergeleken met de theoretische waarden uit het handboek. Voor dit labo werk je in groepen van 2 personen.

## 2 Opdracht

De opdracht bestaat uit drie belangrijke stappen. De eerste stap is de modellering van het probleem en het inlezen van de XML-dataset. Wanneer je deze stap gemaakt hebt kan je starten met het implementeren van de verschillende strategieën uit het handboek. Tot slot ga je na of deze waarden overeenstemmen met de waarden uit het handboek en verklaar je de verschillen tussen de strategieën.

### 2.1 XML-dataset

De dataset is te vinden op Toledo. Deze dataset bevat een lijst van processen. De aankomsttijden van de processen zijn Poisson verdeeld met een gemiddelde rate van 0.8. De bedieningstijden zijn exponentieel verdeeld met een rate van 1.0. Per proces is volgende informatie gegeven:

- `<pid>` de PID van het proces
- `<arrivaltime>` de aankomsttijd van het proces in aantal JIFFY's
- `<servicetime>` de bedieningstijd van het proces in aantal JIFFY's

De tijden zijn niet in seconden, maar in aantal *JIFFY*'s. Een JIFFY is de tijd tussen twee "system timer"-interrupts. Er kan enkel gescheduled worden, beslissen welk proces aan de beurt is, bij een interrupt. Bij de dataset die je krijgt is de waarde van de JIFFY, 10 ms.

### 2.2 Processtrategieën implementeren

Wanneer de modellering en het inlezen van de dataset voltooid is kan je de processen scheduler volgens de strategieën uit het handboek. Je implementeert daarbij de volgende algoritmen: FCFS, SJF, SRT, RR, HRRN, en multilevel feedback mode. Van het FCFS, SJF, SRT en HRRN algoritme maak je 1 versie. Het RR algoritme wordt geïmplementeerd met time slices  $q = 2$ ,  $q = 4$  en  $q = 8$ . Voor het multilevel feedback algoritme maak je twee versies, telkens met 5 wachtrijen. Maak hier zelf een goeie keuze voor de time slices per wachtrij. Motiveer de keuze in het verslag. Uiteraard mag je er meer implementeren dan deze. Schenk bij de implementatie van deze strategieën aandacht aan het ontwerp. Indien de modellering goed verlopen is, zal je minder problemen hebben in deze fase.

### 2.3 Evaluatie testresultaten

Voor het evalueren van de scheduler bepaal je aan de hand van een testrun een aantal grootheden:

- **per proces:** aankomsttijd, bedieningstijd, starttijd, eindtijd, omlooptijd, genormaliseerde omlooptijd en wachttijd
- **globale parameters:** gemiddelde omlooptijd, gemiddelde genormaliseerde omlooptijd en gemiddelde wachttijd

De resultaten breng je in twee grafieken:

- genormaliseerde omlooptijd in functie van bedieningstijd
- wachttijd in functie van bedieningstijd

De bedieningstijd verdeel je in percentielen. Voorbeeld voor 20000 processen: het eerste percentiel bestaat uit de 200 kortste bedieningstijden.

## 3 Rapportering

Het rapport – exact 6 paginas – bevat de volgende onderdelen:

- *Modellering en structuur van het programma (1 pagina).* Neem hier géén broncode op.
- *Testresultaten en evaluatie (4 paginas).* Je neemt 4 grafieken op in je evaluatie, namelijk (1) de genormaliseerde omlooptijd in functie van de bedieningstijd voor 10000 processen, (2) de genormaliseerde omlooptijd in functie van de bedieningstijd voor 20000 processen, (3) de wachttijd in functie van de bedieningstijd voor 10000 processen en (4) de wachttijd in functie van de bedieningstijd voor 20000 processen. Elke grafiek wordt op een halve pagina weergegeven. In totaal neemt dit dus 2 paginas in beslag. Daarnaast evalueer en interpreteer je de grafieken in 2 pagina's.
- *Besluit, reflectie en tijdsbesteding (1 pagina).* Bij de reflectie licht je kort toe wat je wel/niet bijgeleerd hebt, wat er goed/fout gelopen is, hoe de samenwerking in de groep verliep.

## 4 Evaluatie

Het geleverde werk wordt gequoteerd op 50 procent van de labopunten en bestaat uit 3 componenten:

- De kwaliteit van het rapport dat wordt ingediend. Hierbij is speciale aandacht voor de evaluatie en vergelijking van de testresultaten.
- De kwaliteit van de code en de demonstratie tijdens de mondelinge presentatie, evenals de correctheid van de geïmplementeerde strategieën.
- De mondelinge presentatie die wordt gegeven, en het beantwoorden van de vragen tijdens deze voorstelling.

## 5 Praktische informatie

Het rapport en de broncode mail je naar [vincent.naessens@cs.kuleuven.be](mailto:vincent.naessens@cs.kuleuven.be) **uiterlijk 31 maart 2021** om 8u des morgens.