

Practicum 4

De scheduler – deel 2

In dit practicum passen we de scheduler verder aan. Het is noodzakelijk dat je voor de aanvang van deze opgave het eerste deel hebt opgelost. De inleidende tekst vindt je ook daar terug op pagina ??

4.1 Indienen

Dit practicum wordt gequoteerd. Je moet zowel de verschillende versies van je code als een verslag indienen voor **24 april, 22:00:00** via <https://indiano.ugent.be>. Let op! We verwachten per groep één enkel zip bestand waarin de volgende bestanden zijn opgenomen.

- practicum4_deel2_verslag.pdf
- practicum4_deel2_vraag1.asm
- practicum4_deel2_vraag2.asm
- practicum4_deel2_vraag3.asm
- practicum4_deel2_vraag5.asm

Zoals steeds gelden de algemene richtlijnen voor het indienen van verslagen.

4.2 Opgaven

1. Schrijf de routine creëertaak die een nieuwe taak aan de takenlijst toevoegt en die drie argumenten heeft:
 - (a) Het adres van de eerste instructie die moet uitgevoerd worden.
 - (b) Het adres van de stapel die door deze taak gebruikt kan worden.
 - (c) De tijd waarop gewacht moet worden alvorens de taak mag starten. Dit is dus de waarde die Huidige_Tick moet bereiken alvorens de taak mag gestart worden.

Deze creëertaak wordt opgeroepen als creëertaak(adres, stapel, wachttijd), en moet:

- (a) De stapel van de nieuwe taak initialiseren zodat het lijkt alsof de taak onderbroken werd net voor de uitvoering van de eerste instructie in die taak. Op die manier is de taak klaar is om gestart te worden door de schedulerhandler.

- (b) Op een lege plaats in de takenlijst de resulterende stapelwijzer (dus na het goedzetten van de stapel!) en kloktik invullen. Die kloktik is de tick waarna de taak mag beginnen uitvoeren.
- (c) Ervoor zorgen dat de IF-bit op 1 staat in de data die de EFLAGS voorstellen.

Hierbij willen we ook weten:

- (a) Waarom moet de IF-bit op 1 gezet worden?
- (b) Waarom moet je deze vlaggen op de nieuwe stapel bewaren?

Tip: Maak hierbij gebruik van de informatie die je in de voorbereidings-opgaven verzameld hebt.

Om het geheel uit te testen reduceer je de spiraal uit het hoofdprogramma tot de rechterhelft van het scherm (b.v. 40-79), en creëer je een linkerspiraal die de linkerhelft voor zijn rekening neemt (b.v. Taak1 uit de inleiding). Start deze taak op met de routine creeertaak en zorg ervoor dat deze taak pas later gestart wordt (b.v. na 2500 kloktikken; kies dit getal niet te klein, anders gebeurt dit opstarten zo snel na het booten dat je niet zal merken of de taak al dan niet correct wacht met starten). Gebruik steeds stapel1 voor taak1, stapel2 voor taak2, ...

Sla je bestand op als `practicum4_deel2_vraag1.asm`.

2. Zorg er nu voor dat het hoofdprogramma geen spiraal meer uitvoert, maar enkel nog de twee taken Taak1 en Taak2 uit de inleiding opstart. Laat het hoofdprogramma nadien wachten in een eindeloze lus (die we later zullen wegwerken). Start ook meteen de informatieve taak PrintInfoTaak, die informatie over de scheduling op het scherm zal printen.

PrintInfoTaak is een taak die informatie afprint die handig kan zijn bij het debuggen van je code. Wat je hieruit kan afleiden is:

- (a) Of de ticks nog verhogen
- (b) Indien de ticks niet verhogen, kan het zijn dat er geen schedulingswitches meer gebeuren. Indien de spiralen niet meer draaien, maar het timestamp counter (TSC) veldje verhoogt, dan weet je dat deze infotaak wel actief is.
- (c) Bij elk taakslot zie je tot welke clocktick deze taak zal slapen, en of er wel een taak in zit (de 'A' staat hier voor *Actief*, de 'T' voor *geTermineerd*).

Sla je bestand op als `practicum4_deel2_vraag2.asm`.

3. We willen ook taken kunnen beëindigen, zodat deze niet langer gescheduled worden. Implementeer hiertoe de routine termineertaak.

Deze routine moet het element dat verwijst naar de stapel van de taak behorende bij de Huidige_Taak in de takenlijst op 0 zetten, zoeken naar een andere taak die mag uitgevoerd worden. Nu kan je naar believen taken opstarten en laten termineren.

Sla je bestand op als `practicum4_deel2_vraag3.asm`.

4. Op dit punt kunnen we zonder problemen de oneindige lus in de hoofdtak wegwerken door termineertaak op te roepen in plaats van de oneindige lus uit te voeren. Als je de vorige stappen correct hebt uitgevoerd, zal dit werken.

Je kan dit eenvoudig verifiëren door naar de uitvoer van de informatietaak te kijken: het slot van de main-functie waarbij voorheen een 'A' werd geprint, zal nu een 'T' bevatten.

5. We willen er nu voor zorgen dat de scheduler aangeeft wanneer er geen taken kunnen uitgevoerd worden. Zorg ervoor dat in dat geval een boodschap geprint wordt, bijvoorbeeld `GEEN TAKEN`. Om dit te testen moet je de spiraal termineren na X ticks, bijvoorbeeld $X=5000$. Je taken termineer je door in de 'herstart' functionaliteit van spiraal op de aangeduide plaats te controleren of `Huidige_Tick` deze vooraf bepaalde waarde overschrijft, en zo nodig termineertaak op te roepen. (Uiteraard ga je dan voor deze opgave dan ofwel de `InfoTaak` niet opstarten, of ga je ze ook moeten termineren na X ticks.)

Sla je bestand op als `practicum4_deel2_vraag5.asm`.

Succes!