

## Operating Systems

### Opdracht 2

#### 4.1

- Wanneer we een array van 0 tot  $n$  elementen willen optellen kunnen we die array splitsen in een array van 0 tot  $n/2$  en  $n/2$  tot  $n$ . Als we dan met twee threads deze rijtjes optellen en daarna de rijtjes bij elkaar tellen is dit sneller dan alles achtereenvolgens op te tellen.
- Het is handig om threads te gebruiken bij programma's die vast lopen wanneer 1 thread gebruikt wordt, bijvoorbeeld een browser. Hierbij kan een thread gemaakt worden om data van internet af te halen en om de interface te laten zien. Hierdoor wordt de responsiveness van het programma groter.

#### 4.3

Heap memory en global variables worden gedeeld.

#### 4.4

Ja, als er meerdere threads aangemaakt worden op een single-processor system worden ze effectief nog steeds na elkaar uitgevoerd. Als er meerdere threads aangemaakt worden op een multi-processor system kunnen de threads in verschillende processorkernen tegelijk worden uitgevoerd.

#### 4.6

Ja, concurrency betekent dat een systeem meerdere processen uit kan voeren. Parallelism betekent dat een systeem meerdere dingen tegelijk uit kan voeren. Door snel te switchen tussen processen kan concurrency bereikt worden. Het is dus mogelijk om concurrency te hebben op een systeem zonder dat het systeem Parallelism heeft.

#### 4.8

- The multithreaded statistical program described in Exercise 4.21
  - Task parallelism
- The multithreaded Sudoku validator described in Project 1 in this chapter
  - Task parallelism
- The multithreaded sorting program described in Project 2 in this chapter
  - Data parallelism
- The multithreaded web server described in Section 4.1
  - Task parallelism

4.10

Main

Child

Child

Thread

Child

Thread

Child

Child

Main maakt twee childs aan, het eerste child van main maakt een child, een thread en nog een child aan. Het eerste Child maakt ook een thread en Child aan. In totaal is dat dus:

- a. 5 nieuwe Processen
- b. 2 nieuwe Threads