

# ICT Infrastructuuren: Processen en Threads

18 november 2013

David N. Jansen

# Datum en tijd van werkcollege

- na overleg met de aanwezigen:  
donderdag 8:45–10:30

# Leerdoel voor vandaag



- Hoofddoelen en -functies van besturingssystemen
- Processen
- Threads
  
- Lees het boek voor de details!

# In- en uitvoer

- Communicatie met de buitenwereld
- Maakt gebruik van de bus
- Drie manieren om te communiceren
  - Programma-gestuurd
  - Interrupt-gestuurd
  - DMA, direct memory access

# Direct Memory Access

- Als er invoer is, wacht de processor even
- invoerapparaat mag de bus gebruiken om gegevens direct naar RAM te schrijven

# Leerdoel voor vandaag



- Hoofddoelen en -functies van besturingssystemen
- Processen
- Threads
  
- Lees het boek voor de details!

# Doelen van Besturingssystemen

- het OS bemiddelt tussen gewone programma's en de hardware
- gebruiksgemak (voor de programmeur)
- efficiëntie
- ontwikkeling (van nieuwe hardware e.d.) faciliteren

# Bemiddeling

- processor beschouwt OS als een soort programma
  - kernel = centraal deel van OS (altijd in RAM)
- gebruiker/andere programma's beschouwen OS (bijna) als deel van hardware
  - OS beheert de hardware



# Diensten voor gewone programma's

- programma's ontwikkelen
- programma's starten
- abstractie van hardware (bronnen)
  - toegang tot in- en uitvoer
  - toegang tot bestanden
  - foutafhandeling
- gebruikersbeheer
  - toegangsrechten
  - boekhouding

# Beheer van bronnen (resources)

- de computer bestaat uit bronnen
  - processor
  - hoofdgeheugen (RAM)
  - secundair geheugen
  - in- en uitvoer van gegevens
- het OS bepaalt  
wie die bronnen mag gebruiken

# Evolutie van Besturingssystemen

Laat de computer niet wachten.

Rekentijd is duur.



Laat de gebruiker niet wachten.

Werktijd is duur.

# Evolutie van Besturingssystemen

- ① handbediening / geen OS
- ② batch processing
  - single
  - multiprogrammed
- ③ interactief timesharing

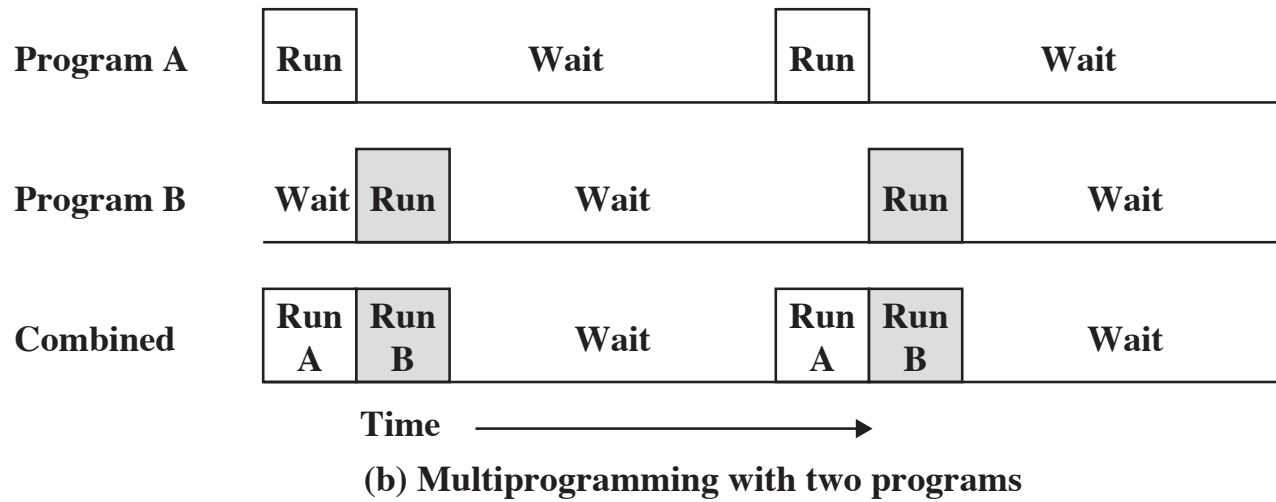
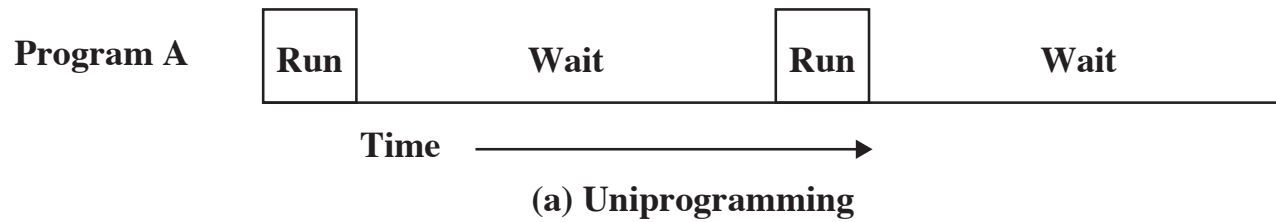
# (Single) Batch processing

- gebruiker levert een stapel ponskaarten in
- beheerders voegen wat extra kaarten toe (Job Control Language)
  - markeren begin en einde van de stapel
  - geven de soort van programma aan
- „besturingssysteem” leest JCL-kaarten en beslist wat te doen
- OS bewaakt correctheid en voortgang

# Multiprogrammed batch processing

- in- en uitvoer is langzaam = veel wachttijd
- als een job wacht, kan een andere draaien
- betere afscherming tussen jobs vereist
  - ingewikkelder geheugenbeheer
  - verdeling van processortijd
  - gelijktijdig gebruik van in- en uitvoer
  - elke job denkt dat hij alleen is op een single-batch-processing-systeem

# Multiprogrammed batch processing



# Interrupts



- doel: bepaalde taken uitvoeren op een gunstig moment
- de processor onderbreekt het lopende programma
- hij voert tussendoor de “interrupt handler” uit (= een ander programma)



# Soorten interrupts

- fout in programma (b.v. deling door 0)
- timer
- in- of uitvoer
- hardwareprobleem

# Voordelen van interrupts

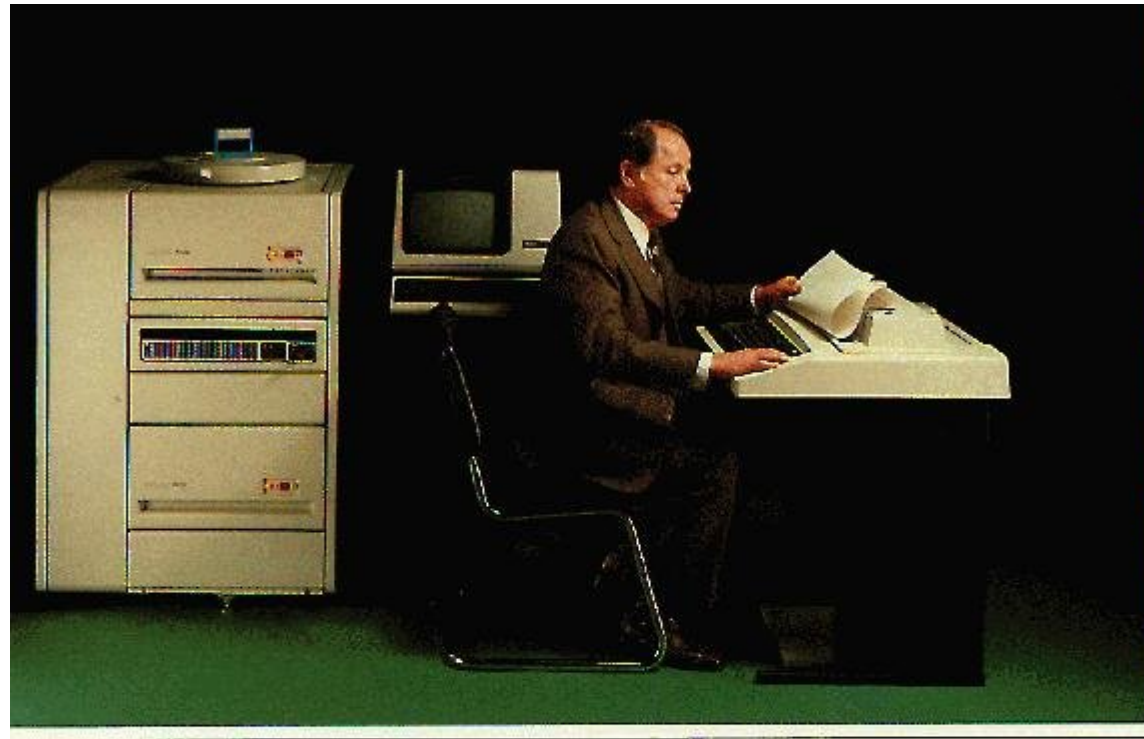
- minder lang wachten op resultaten van in-/uitvoer → CPU wordt beter benut
- exactere timing

# Meerdere interrupts?

- Wat moet er gebeuren?
  - tweede interrupt negeren
  - uitstellen
  - onmiddellijk afhandelen
- interrupt mask/prioriteiten bepalen de keuze
- elke processor lost het probleem een beetje anders op

# Interactief timesharing

- één interactieve centrale computer met meerdere terminals



# Interactief en één gebruiker

- achteruitgang of vooruitgang?
- iedere gebruiker zijn eigen computer
- MS-DOS: uitgekleeed OS
  - geen multiprogramming/multitasking
  - geen gebruikersbeheer

# netwerk van computers

- typische doorontwikkeling: file- en printserver (voor gemeenschappelijke bronnen)
- multitasking en toegangsrechten weer toegevoegd
- internet: netwerk van netwerken

# Proces

- belangrijkste thema vandaag
- hoofdstukken 3+4 in Stallings' boek
- relevant voor multiprogramming en nieuwere OS-types

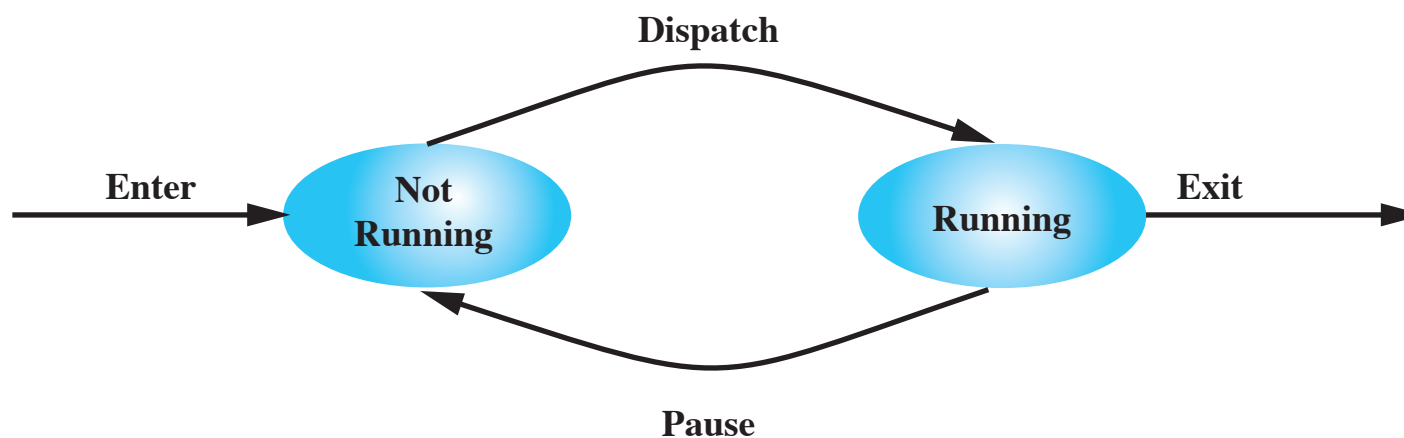
# Wat is een proces?

- een programma dat wordt uitgevoerd onder beheer van het besturingssysteem
- een samenstel van
  - activiteiten
  - status van die activiteiten
    - maakt onderbreken/multiprogramming mogelijk
  - bronnen
- Het OS laat het proces denken dat het alleen is



# Toestand van een proces

- besturingssysteem houdt toestand bij
- één proces is actief (running), de andere processen wachten



(a) State transition diagram

Op weg naar een

# completer toestandenmodel

- Proces ontstaat en eindigt
- Proces wacht om specifieke reden
- Veel processen... te veel voor het RAM

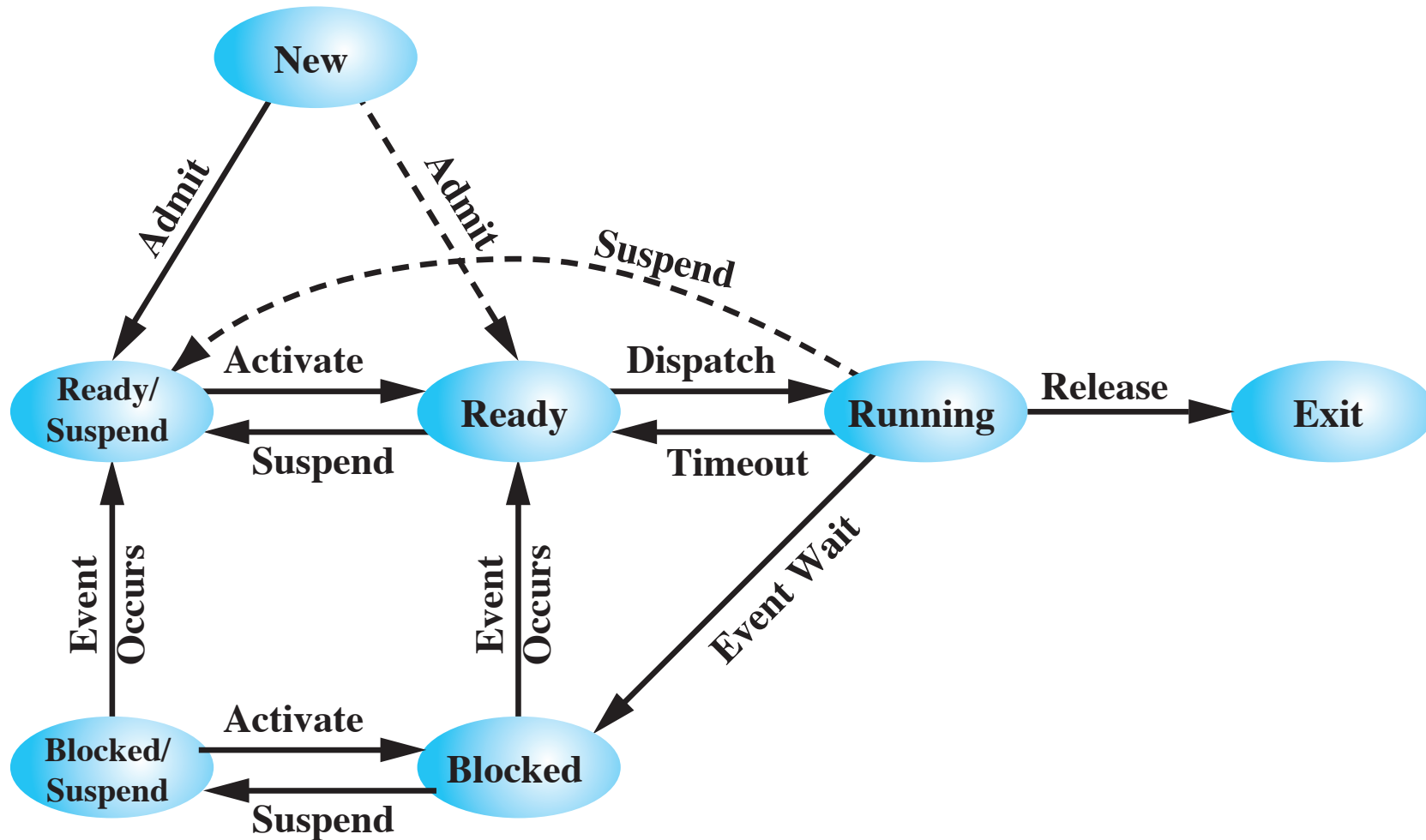
# Proces ontstaat en eindigt

- Proces ontstaat
  - login
  - OS biedt dienst in de vorm van proces aan
  - proces start een nieuw proces
- Proces eindigt
  - normaal einde
  - niet voldoende bronnen
  - proces doet iets illegaals
  - in- of uitvoer kan niet
  - buitenstaander verzoekt om einde

# Redenen voor wachten

- proces is niet aan de beurt
- in- of uitvoer duurt langer
- buitenstaander onderbreekt het proces
  
- niet voldoende RAM:  
proces wordt op harde schijf opgeslagen

# Completer toestandenmodel



(b) With Two Suspend States

# Informatie over het proces

- Het besturingsstelsel houdt bij:
  - identiteit van het proces
  - processor-registers en geheugen
  - toestand van het proces (actief / blocked / ...)
  - bronnen die het proces gebruikt
  - details in tabellen 3.4 en 3.5 (p. 148vv.)
- Process Control Block bevat deze informatie

# Hardware-ondersteuning voor OS

- instructies met privileges
- user mode: voor gewoon programma  
sommige instructies verboden
- kernel mode: voor OS  
alle instructies toegestaan

# Mode switch

- direct omschakelen user → kernel mode is verboden
  - omschakelen alleen tegelijk met
    - interrupt of trap
    - system call = programma roept functie van OS aan
- direct omschakelen kernel → user mode is toegestaan
  - automatisch aan het einde van een interrupt



# Process switch =

## Omschakelen tussen processen

- taak van het OS
  - 1 oude context opslaan
    - context = toestand van CPU (registers) en RAM
  - 2 kiezen welk proces aan de beurt is
  - 3 procestoestanden veranderen
  - 4 nieuwe context laden
  - 5 naar juiste instructie in nieuw proces springen

# Threads

- een lichte variant van processen
- processen gebruiken twee soorten bron:
  - processortoegang
  - geheugen, bestanden etc.
- threads gebruiken slechts één bron:
  - processortoegang
- eenvoudiger dan proces
  - ⇒ omschakelen is sneller

# Proces vs. thread

- Procesinformatie
  - geheugen
  - bronnen (bestanden, in- en uitvoer)
- Threadinformatie
  - toestand van de thread (running / blocked / ...)
  - programmateller
  - lokale variabelen (“stack”)

# Process vs. thread

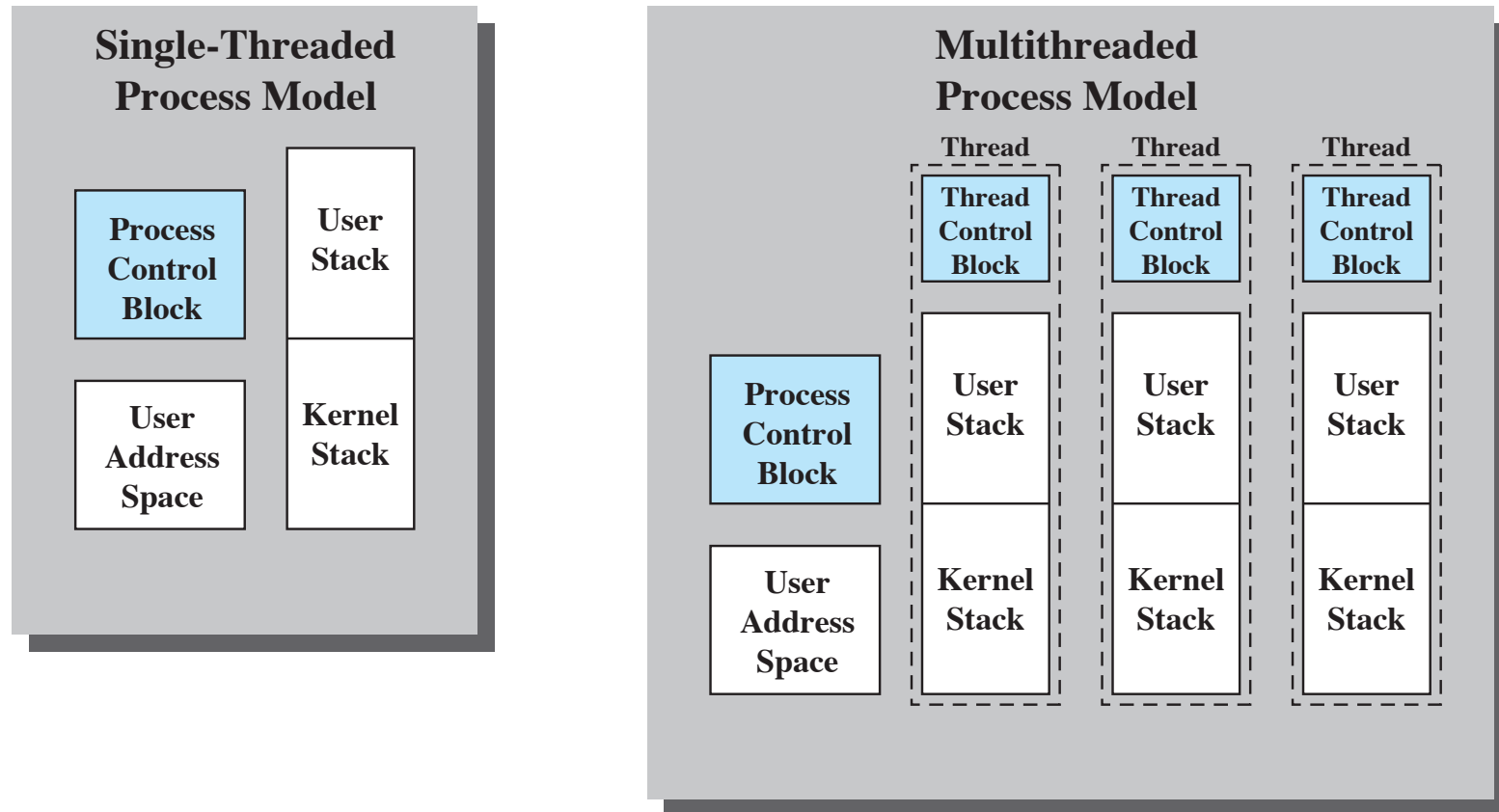


Figure 4.2 Single Threaded and Multithreaded Process Models

# Waarom dienen threads?

- verschillende taken in één programma
  - reageren op gebruiker
  - commando's uitvoeren
  - achtergrondtaken
- sommige taken kunnen doorgaan terwijl het programma op OS wacht
- multicore

# OS-ondersteuning

- Sommige OSen bieden geen threads aan
  - dan zijn toch “user level threads” mogelijk
- wel ondersteuning: “kernel level threads”
  - langzamer, maar flexibeler

# Symmetric Multiprocessing

- vroeger: 1 processor per applicatie
- nu: multicore
- processoren zijn symmetrisch  
= kunnen dezelfde instructies etc. aan

# Opbouw van een SMP-systeem

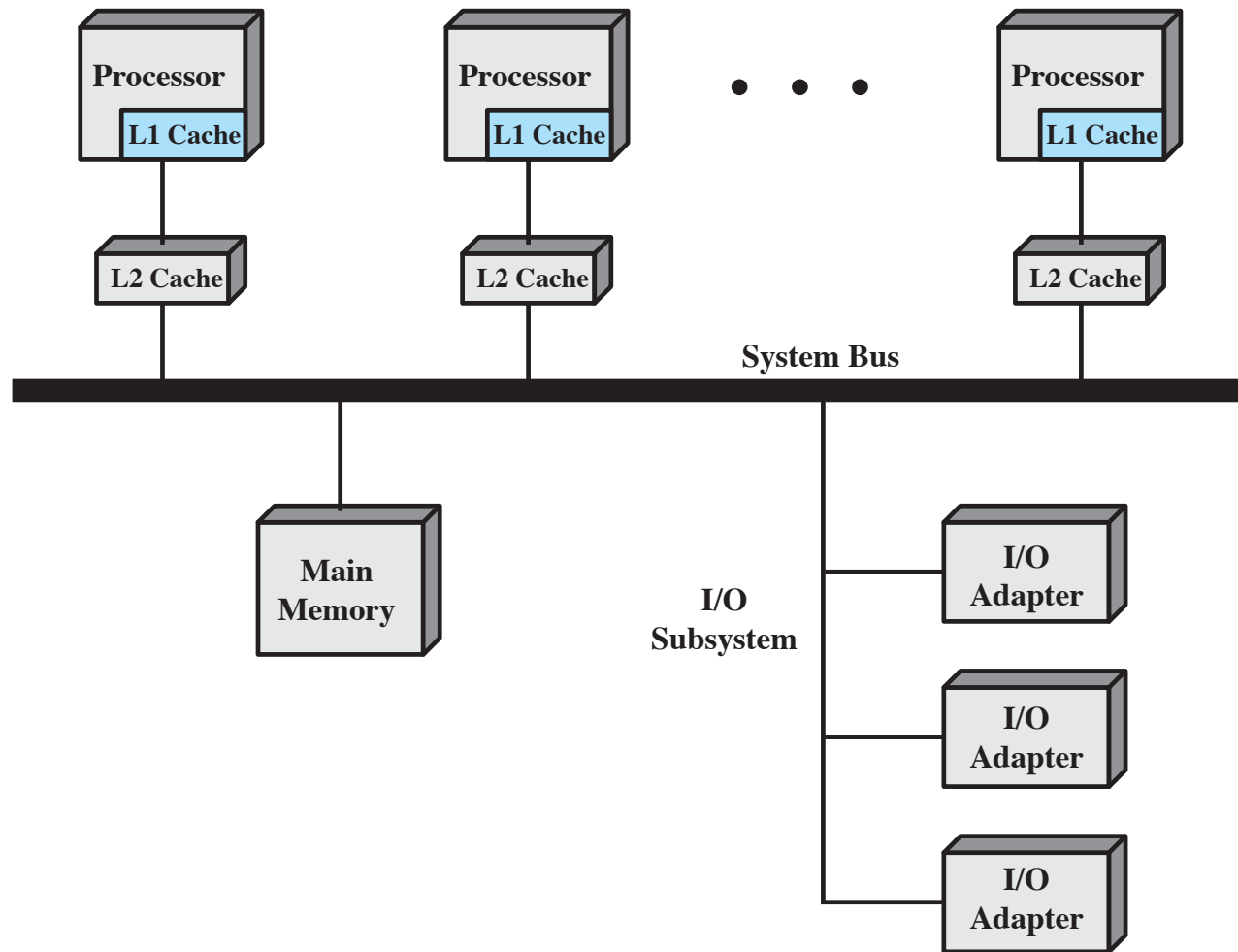


Figure 4.9 Symmetric Multiprocessor Organization



# Problemen van SMP

- meerdere processoren executeren zelfde code  
→ code moet daarmee rekening houden
- vaak is serialisatie gewenst  
(het lijkt alsof er alleen 1 processor was)
- meerdere system calls tegelijk mogelijk  
→ mutual exclusion is ingewikkelder
- wie verdeelt de processen?

# Samenvatting

- Doelen van besturingssysteem
- Functie, service van besturingssysteem
- Proces
  - een programma dat draait onder beheer van een besturingssysteem
- Thread
  - lichtgewicht-proces
  - met gemeenschappelijk geheugen en bronnen