



HOGESCHOOL ROTTERDAM / CMI

Besturingssystemen 2

TINBES02-2

Aantal studiepunten: 3

Cursusbeheerder: dr. W. M. Bergmann Tiest



Cursusbeschrijving

Cursusnaam:	Besturingssystemen 2
Cursuscode:	TINBES02-2
Aantal studiepunten en studiebelastinguren:	3 ec Dit studieonderdeel levert de student 3 ec op, hetgeen overeenkomt met een studielast van 84 uren waarvan 10 contacturen. De verdeling van deze 84 uren is als volgt: <ul style="list-style-type: none"> • 7 × 45 minuten on-line hoorcollege: 5 uur. • 7 × 45 minuten on-line werkcollege: 5 uur. • zelfstudie en onbegeleid werken aan de eindopdracht: 73 uur. • demonstreren eindopdracht: 1 uur.
Vereiste voorkennis:	<ul style="list-style-type: none"> • Computersystemen & Logica (TINCPS02-1) • Programmeren 1 (TINPRO02-1) • Besturingssystemen 1 (TINBES02-1) • Programmeren 5B (TINPRO025B)
Werkvorm:	On-line hoor- en werkcollege.
Toetsing:	Opdracht.
Leermiddelen:	<ul style="list-style-type: none"> • Arduino Uno of Nano met USB-kabel. • Boeken (niet verplicht): Andrew S. Tanenbaum & Herbert Bos: <i>Modern Operating Systems</i>, 4th edition (2015). Brian W. Kernighan & Dennis M. Ritchie: <i>The C Programming Language</i>, 2nd edition (1988). • Slides worden op Teams geplaatst voor de start van de les.
Leerdoelen:	<p>De student:</p> <ol style="list-style-type: none"> 1. Weet hoe een besturingssysteem werkt. 2. Kan een simpel besturingssysteem implementeren en testen. 3. Begrijpt hoe input/output werkt en kan dit implementeren voor een besturingssysteem. 4. Begrijpt hoe dynamisch geheugen werkt en kan dit implementeren voor een besturingssysteem. 5. Begrijpt hoe multi-tasking werkt en kan dit implementeren voor een besturingssysteem. 6. Begrijpt hoe een file system werkt en kan dit implementeren voor een besturingssysteem.
Inhoud:	Tijdens deze cursus ga je zelf een besturingssysteem implementeren voor de Arduino Uno, inclusief geheugenbeheer, bestandssysteem, input/output (command line interface) en multitasking. Hiermee kun je meerdere programma's tegelijkertijd laten draaien op de Arduino, en data opslaan in bestanden. Daarnaast worden geavanceerde concepten behandeld die betrekking hebben op besturingssystemen, zoals semaforen, shared memory, multitasking en prioritisering.
Opmerkingen:	De opdracht moet individueel worden uitgevoerd.
Cursusbeheerder:	dr. W. M. Bergmann Tiest
Datum:	9 april 2021

1 Algemene omschrijving

1.1 Inleiding

Een besturingssysteem (OS: Operating System) is de belangrijkste software die op een computer wordt uitgevoerd. Het OS beheert alle software en hardware op de computer. Het regelt het geheugenbeheer, bestandsbeheer, de input/output, zorgt voor interrupts, voor het uitvoeren van meerdere onafhankelijke computerprogramma's op dezelfde computer, enzovoort. Zonder een OS is een computer nutteloos. Meestal zijn er verschillende computerprogramma's tegelijkertijd actief en ze moeten allemaal toegang hebben tot de CPU, het geheugen en de opslag van de computer. Het OS coördineert taken om ervoor te zorgen dat elk programma krijgt wat het nodig heeft.

De drie meest gebruikte OS'en zijn Microsoft Windows, MacOS en Linux. In Besturingssystemen 1 heb je kennis gemaakt met het Linux OS, en enkele onderdelen van een OS geïmplementeerd. Besturingssystemen 2 bouwt hierop voort. Tijdens deze cursus ga je zelf een besturingssysteem implementeren voor de Arduino Uno, inclusief geheugenbeheer, bestandssysteem, input/output (command line interface) en multitasking. Hiermee kun je meerdere programma's tegelijkertijd laten draaien op de Arduino, en data opslaan in bestanden. Daarnaast worden geavanceerde concepten behandeld die betrekking hebben op besturingssystemen, zoals semaforen, shared memory, multitasking en prioritisering.

1.2 Relatie met andere onderwijseenheden

Besturingssystemen 2 gebruikt de kennis die de studenten tijdens de cursus Besturingssystemen 1, Computersystemen en de programmeer-leerlijn hebben opgedaan. Het begrip van de interne werking van het besturingssysteem ligt aan de basis van het gebruik van computers in de projecten, stage, TINLab en afstuderen.

1.3 Leermiddelen

- Arduino Uno of Nano met USB-kabel.
- Boeken (niet verplicht):
Andrew S. Tanenbaum & Herbert Bos: *Modern Operating Systems*, 4th edition (2015).
Brian W. Kernighan & Dennis M. Ritchie: *The C Programming Language*, 2nd edition (1988).
- Slides worden op Teams geplaatst voor de start van de les.

2 Programma

Tijdens de lessen wordt je begeleid bij het bouwen van een eigen besturingssysteem voor de Arduino Uno. De specificaties en eisen hiervoor worden in een apart document meegedeeld. Het besturingssysteem wordt geschreven in C; hiervoor wordt eerst een aantal geavanceerde constructies herhaald, zoals pointers en structs. Voor het bestandssysteem wordt gebruik gemaakt van het EEPROM van de Arduino; dit wordt ook uitgelegd. Daarna worden geavanceerde concepten behandeld die betrekking hebben op besturingssystemen, zoals semaforen, shared memory, multitasking en prioritisering.

Er zijn twee on-line lessen per week: één les met alle deelnemers aan de cursus waarin de theorie behandeld wordt en uitleg gegeven wordt over hoe de opdracht aan te pakken, en één les met alleen de eigen stamgroep waarin de stof verder uitgediept wordt, waarin programmeerproblemen besproken kunnen worden, en waarin de docent beschikbaar is voor consultancy. Buiten de lessen moet op eigen gelegenheid aan het OS gewerkt worden.

Voor de ontwikkeling van het OS is er een aantal *checkpoints*: stadia in de ontwikkeling die gevolgd moeten worden. We raden je nadrukkelijk aan om te zorgen dat aan het eind van iedere week het bijbehorende checkpoint behaald is, want als je eenmaal achterloopt is inhalen erg moeilijk. Je bent als student hier *zelf* voor verantwoordelijk.

Week	Lesinhoud	Checkpoint OS
1	Herhaling C programmeren: pointers, structs; uitleg command line	Basisstructuur & ontwerp
2	BIOS, EEPROM & PROGEM; uitleg bestandssysteem	Command line interface
3	Shared memory & semaforen; uitleg memory & process table	Bestandssysteem
4	Geen les (2 ^e pinksterdag)	Geheugenbeheer en variabelen
5	Multitasking & prioritisering; uitleg stack	Processen
6	Uitleg instructies uitvoeren: basisinstructies	Stack
7	Uitleg instructies uitvoeren: flow control	Instructies uitvoeren
toetsweek	Demonstratie opdracht	

3 Toetsing en beoordeling

3.1 Procedure

De toetsing bestaat uit het demonstreren van de eindopdracht aan de docent en het inleveren van de broncode. Tijdens de demonstratie wordt de student ondervraagd over de werking van het programma en toont de student aan op welke manier aan de leerdoelen voldaan is.

3.2 Beoordeling

Voor ieder van de volgende aspecten wordt 0, 0.5 of 1 punt gegeven, als aan het aspect respectievelijk niet of nauwelijks, voor ongeveer de helft, of geheel is voldaan. De items 10–13 zijn bonusitems; deze zijn niet in de practicumhandleiding beschreven en bieden extra uitdaging. Voor deze bonusitems worden alleen punten toegekend als er met de basisitems 1–9 minstens 6 punten behaald zijn. Het eindcijfer is de som van de punten voor de verschillende aspecten. Het maximale eindcijfer is 10. Tussen haakjes staan de betreffende leerdoelen (zie p.1).

1. Het OS leest opdrachten die op de command line worden gegeven (2, 3).
2. Vanuit de command line kan data in bestanden worden opgeslagen, een lijst getoond van opgeslagen bestanden, de inhoud van bestanden getoond, bestanden gewist en de hoeveelheid beschikbare vrije ruimte getoond (1, 2, 6).
3. Er kunnen variabelen in het geheugen opgeslagen worden van de typen CHAR, INT, FLOAT en STRING, weer teruggehaald uit het geheugen, en van waarde veranderd (1, 2, 4).
4. Een bestand in het bestandssysteem kan als proces gestart, gepauzeerd en gestopt worden (1, 2).
5. Er wordt van verschillende processen bijgehouden wat de toestand is (running, suspended, terminated), de waarde van de program counter, en welke variabelen bij dat proces horen (1, 2).
6. Het OS kan instructies in een bytecode-programma uitvoeren (1, 2).
7. Het OS kan meerdere processen tegelijk uitvoeren (afwisselend één instructie van elk proces) (1, 2, 5).
8. Vanuit een bytecode-programma kan data van de typen CHAR, INT, FLOAT en STRING naar bestanden geschreven worden en uit bestanden gelezen (1, 2, 6).
9. Vanuit een bytecode-programma kunnen andere processen opgestart (geforkt) worden en kan gewacht worden op het voltooien van een ander proces (1, 2, 5).
10. **Bonus:** Er zijn faciliteiten in het OS aanwezig voor het toekennen van prioriteiten aan processen, en een proces wordt uitgevoerd met de toegekende prioriteit (5).
11. **Bonus:** Er zijn faciliteiten in het OS aanwezig voor het uitwisselen van data tussen twee processen door middel van shared memory (5).
12. **Bonus:** Er zijn faciliteiten in het OS aanwezig voor het gebruik van semaforen voor gedeelde bestanden of shared memory (5).
13. **Bonus:** Er is een compiler aanwezig die (een variant van) C- of Python-code omzet naar bytecode geschikt voor het OS (hoeft niet op Arduino te draaien; parser hoeft niet zelfgeschreven te zijn) (1, 2).

3.3 Herkansing

De herkansing bestaat uit het opnieuw inleveren en demonstreren van de eindopdracht uiterlijk tijdens uitloopweek 1 van OP4. Hogerejaarsstudenten die het vak nog moeten halen, moeten een afspraak maken met de docent voor het inleveren en demonstreren van de opdracht.

4 Aanwezigheid

Aanwezigheid is verplicht bij de mondelinge toelichting van de eindopdracht, omdat hiermee vastgesteld wordt of de student eigen werk heeft ingeleverd.

5 Changelog

Studiejaar	Wijzigingen
2017–2018	eerste versie
2018–2019	wordprocessor-opdracht toegevoegd beoordeling verduidelijkt
2019–2020	opdrachten geïntegreerd tot 1 eindopdracht semaforen, shared memory en multitasking toegevoegd beoordeling herzien
2020–2021	aangepast aan on-line onderwijs afvinken checkpoints eruit gehaald bonusitems toegevoegd aan beoordeling

Bijlage: Toetsmatrijs

Leerdoel	Bloom-niveau
1	2
2	3
3	2, 6
4	2, 6
5	2, 6
6	2, 6

Bloom-niveaus:

1. Remember
2. Understand
3. Apply
4. Analyse
5. Evaluate
6. Create