M E M O R A N D U M

Universiteit van Stellenbosch
Departement Rekenaarwetenskap

Kursus: RW 314

April 2002 Tydsduur: $2\frac{1}{2}$ uur

Algemene Konsepte *General Concepts*

1. Verduidelik kortliks die verskil tussen verspreide- en paralellestelsels.

   Briefly explain the difference between distrubuted and parallel systems.

   > Parallel systems, also called tightly coupled systems, utilizes multiple CPUs in a single machine and share resources such as the bus and system clock between the various CPUs. Distributed systems, also called loosely coupled systems, distribute processing amongst different workstations and relies on communication through high speed networks or telephone lines. The CPUs in a distributed system do not share resources such as the system clock.

   2

2. Verduidelik wat bedoel word met die term **direkte geheuetoegang** (DMA), met spesiale verwysing na die gebruik daarvan in skyfbeheereenhede. Wat is die voordele daarvan om DMA op hierdie manier te gebruik?

   *Explain what is meant by the term **direct memory access** (DMA) with special reference to its use in disk control units. What are the advantages of using DMA in this way?*

   > DMA refers to a mechanism that supports asynchronous I/O. When the operating system receives a request to transfer data to or from an I/O device, it initializes the DMA controller by sending various parameters such as the amount of data and the address of the buffer that must be used during the transfer. Once the DMA controller has been programmed, a command is send to the I/O device to either read or write the data using DMA. Once the command has been send to the device, the operating system can continue by scheduling another process. The I/O device and DMA controller will then exchange the data between themselves without the requiring any intervention from the CPU. When the transfer is complete, the I/O device will generate an interrupt, signalling that the operation has been completed. Although DMA allows the CPU to continue with other processing operations, some CPU cycles are lost (cycle stealing) because there will be contention between the CPU and DMA controller to access the system bus. If the I/O requested access to a disk, the operating system will begin by initializing the DMA controller, before sending either a read or write command to the disk controller. Once the disk controller receives the command, it will begin executing the required operation and exchange the data with the DMA controller until the operation has been completed afterwhich the disk controller will generate an interrupt and control is transfered back to the operating system.

   6

3. Noem en bespreek kortliks twee tipes dienste wat deur 'n bedryfstelsel verskaf word.

   Name and briefly discuss two types of services provided by an operating system.

   - **Communication** Operating systems must provide the means for different processes to communicate with each other, for example shared memory and message passing.
   - **File System** A file system is usually provided to allow users to store and retrieve their data or to store programs that can be loaded at a later stage for execution.

4

Prosesbestuur *Process Management*

4. Bespreek kortliks die verskil tussen 'n program, proses en liggewigproses.

   Briefly explain the difference between a program, process and lightweight process.

   A program is a file containing object code (executable machine instructions) and is usually kept on secondary storage. Whenever a program is loaded into memory and executed, the program becomes a process within the system. A Lightweight process is similar to a normal process, except that it does not require as much information to be managed. The process control block for a lightweight process only requires the CPU registers and program counter to be stored.

3

5. Bespreek kortliks onder watter omstandighede nie-voorspringskedulering sal plaasvind.

   Briefly describe the conditions for non-preemptive scheduling.

   - When a process switches from a running state to a waiting state, for example, by executing a system call.
   - When a process terminates.

2

6. Daar bestaan 'n hele aantal faktore wat as kriteria dien ten einde 'n potensiële skeduleringsalgoritme te ontleed. Noem twee faktore wat as kriteria gebruik kan word en bespreek kortliks die belang daarvan.

   A number of factors exist that define the criteria used to analyze potential scheduling algorithms. Name two factors that can be used as criteria and briefly discuess their importance.

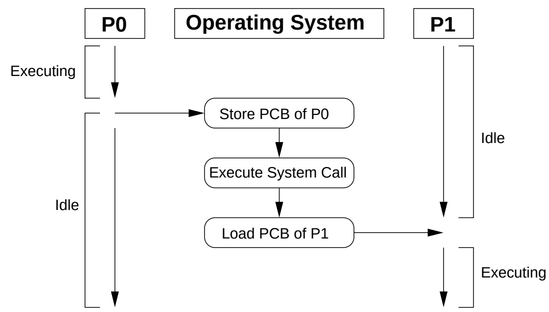
   - **CPU Utilization** An efficient scheduling algorithm will try to keep the CPU occupied at all times to waste as little as possible processing cycles.
   - **Waiting Time** Minimizing the amount of time that a process waits before being scheduled for execution will increase the overall processing within the system and can also affect other conditions such as throughput.

6

7. Daar is tans twee prosesse in 'n stelsel, P_0 en P_1. P_0 is tans besig om uit te voer terwyl P_1 wag om geskeduleer te word. P_0 voer nou 'n stelselroep uit. Verduidelik kortliks wat met die verskillende prosesse gebeur en hoe die bedryfstelsel die stelselroep hanteer. Lig u verduideliking toe met 'n skets.

   A system currently contains two processes, P_0 and P_1. P_0 is currently executing while P_1 is waiting to be scheduled. P_0 executes a system call. Briefly explain what will happen with the two processes and how the operating system will handle the system call. Use a diagram to illustrate your answer.

When P_0 executes a system call, control is transfered back to the operating system and P_0 is suspended and its state stored in a process control block. Depending on the type of the system call, P_0 may be placed in a waiting queue. This will most likely occur if the system call required an I/O operation. If possible, the operating system begins executing the requested system call. The scheduler must select another process to schedule, for example P_1. P_1 is removed from the ready queue and its PCB is loaded into memory, changing its state from ready to running. The scheduler will now invoke the dispatcher to transfer control back to P_1 to resume execution.



5

8. Bespreek kortliks die verskil tussen direkte en indirekte proses kommunikasie.

   Briefly discuss the difference between direct and indirect process communication.

   - **Direct Communication** requires a process that initiates communication to explicityly name the recipient or sender and communication can only take place between two processes and only a single link may be established between two processes.
   - **Indirect Communication** takes place through special mailboxes called ports. Any number of processes can communicate with each other on any number of links as long as the various ports are shared between them.

4

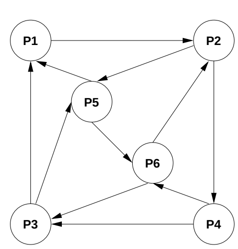
9. Vergrendeling word opgespoor in 'n stelsel deur die toekenning van hulpbronne te ontleed. Die toekenning van 'n hulpbron, R_j, aan 'n proses, P_i, word aangedui as $R_j \to P_i$. Wanneer 'n proses op 'n hulpbron wag word dit aangedui as $P_i \to R_j$. Beskou die onderstaande toekennings en beantwoord die vrae wat volg.

   Deadlock can be detected in a system by analyzing the allocation of resources. The allocation of a resource, R_j, to a process, P_i, is expressed by $R_j \to P_i$. A process waiting to be allocated a resource is expressed by $P_i \to R_j$. Examine the various allocations below and answer the questions that follow.

   $P_1 \to R_1$,$P_2 \to R_5$,$P_2 \to R_7$, $P_3 \to R_3$,$P_3 \to R_4$,$P_4 \to R_8$, $P_4 \to R_9$,$P_5 \to R_2$,$P_5 \to R_{11}$, $P_6 \to R_6$,$P_6 \to R_{10}$,$R_1 \to P_2$, $R_2 \to P_1$,$R_3 \to P_1$,$R_4 \to P_5$, $R_5 \to P_5$,$R_6 \to P_2$,$R_7 \to P_4$, $R_8 \to P_6$,$R_9 \to P_3$,$R_{10} \to P_3$, $R_{11} \to P_6$

   (a) Teken 'n *wait-for* grafiek om aan te dui watter prosesse op mekaar wag.

   Draw a wait-for graph to illustrate which processes are waiting for each other.



3

(b) Wat is die minimum hoeveelheid prosesse wat vernietig kan word gegewe dat die stelsel prosesse individueel verwyder ten einde van vergrendeling te ontsnap? Verskaf die proses, of volgorde van prosesse indien meer as een proses vernietig moet word.
What is the minimum number of processes that must be terminated to resolve the deadlock in the system, given that the system will terminate one process at a time? Write down the process, or sequence of processes if more than one process must be destroyed.

A minimum of 2 processes must be terminated. Any one of the following combinations may be chosen:

- P_2 and P_5
- P_2 and P_6
- P_2 and P_3
- P_4 and P_5
- P_1 and P_5
- P_1 and P_6

2

(c) Die ontwikkelaars van die stelsel glo dat daar so min as moontlik ontwrigting vir hulle gebruikers moet wees. Veronderstel dat die prosesse reeds 'n sekere aantal tydseenhede aan verwerking afgestaan het: P_1 (5), P_2 (4), P_3 (12), P_4 (2), P_5 (11) en P_6 (10). Watter prosesse moet getermineer word ten einde die vergrendeling te breek indien die verlies aan verwerkingstyd 'n minimum moet wees?
The developers of the sytem believe that the user must experience the least possible amount of distruption. Assume that the processes have each spent a certain amount of time executing: P_1 (5), P_2 (4), P_3 (12), P_4 (2), P_5 (11) en P_6 (10). Which processes must be terminated to break the deadlock if the loss in processing time is to be kept at a minimum?

- P_2 and P_5: $4 + 11 = 15$
- P_2 and P_6: $4 + 10 = 14$
- P_2 and P_3: $4 + 12 = 16$
- P_4 and P_5: $2 + 11 = 13$
- P_1 and P_5: $5 + 11 = 16$
- P_1 and P_6: $5 + 10 = 15$

Process P_4 and P_5 must be terminated to minimize the loss in processing time.

2

10. Gestel 'n sekere stelsel het in totaal 12 DVD-RAM aandrywers beskikbaar. Daar bestaan vier prosesse P_0, P_1, P_2 en P_3 wat onderskeidelik 'n maksimum van 6, 5, 4 en 7 DVD-RAM aandrywers benodig.

Assume that a certain system has a total of 12 DVD-RAM drives. There exists four processes P_0, P_1, P_2 and P_3 requiring a maximum of 6, 5, 4 and 7 DVD-RAM drives respectively.

(a) Verduidelik kortliks wat bedoel word met 'n veilige toestand van die stelsel indien die banker's algoritme gebruik word vir die toekenning van DVD-RAM aandrywers.
Explain briefly what is meant by a safe state of the system if the banker's algorithm is used to allocate DVD-RAM drives.

A system is in a safe state if a request to allocate resources can be satisfied and granting this request results in at least one execution sequence of processes that will allow every process to allocate its outstanding resources and safely terminate, releasing its allocated resources, without creating a deadlock.

3

(b) Op 'n sekere tydstip is daar 1 aandrywer toegeken aan P_0, 5 aan P_1, 2 aan P_2 en 4 aan P_3. Is dit 'n veilige toestand? Motiveer u antwoord.
At a given time there is 1 drive allocated to P_0, 5 to P_1, 2 to P_2 and 4 to P_3. Is this a safe state? Motivate your answer.

Yes. The following execution will result in all the processes safely terminating: P_1 has already allocated its maximum need and can complete, releasing 5 instances of the resource. P_0 has an outstanding request for 5 instances and can now allocate, execute and terminate, releasing 6 instances of the resource. The 6 instances of the resource is enough to satisfy the outstanding requests of both P_2 and P_3, allowing them to allocate, execute and terminate.

P_0 (6)	P_1 (5)	P_2 (4)	P_3 (7)	Available
1	5	2	4	0
-	(5)	-	-	5
5	-	-	-	0
(6)	-	-	-	6
-	-	2	-	4
-	-	(4)	-	8
-	-	-	3	5
-	-	-	(7)	12

6

11. Bespreek kortliks die toepassing van monitors en kritiese areas as meganismes om die sinchronisasie van prosesse te verseker.

Briefly discuss the application of monitors and critcal regions as mechanisms to ensure process synchronization.

- Critical regions are high level constructs implemented within a programming language. The data shared between process must be declared as being shared, for example, `VAR x: SHARED LONGINT`. Accessing `x` is only allowed inside a critcal region. Entry to the critical region is controlled by a boolean condition. For example, `REGION x WHEN b DO s` declares a critical region that, when condition `b` is true, may be entered. The statements, `s`, executed in this region may modify the shared variable, `x`. The main advantage of critical regions is that the compiler will generate the necessary code to gain access to the shared variable, for example, using semaphores and takes this responsibility away from the programmer.

- A monitor is also a high level construct that provides access to shared data declared inside the monitor. Access to the shared data occurs through the procedures provided by the monitor. This protects the data from unauthorized access. Synchronization is implemented by using condition variables to protect the data in the monitor when being accessed by multiple processes. Two operations, *signal* and *wait* are provided that control access to the shared data. Whenever a *wait* operation is executed, the process that requested access is placed in a waiting queue. If a *signal* operation is executed, the current process will exit from the monitor and another process is selected from the wainting queue of the condition variable, if any are available. The selected process may now enter the monitor and begin executing.

8

Geheuebestuur *Memory Management*

12. Beskryf kortliks die verwantskap tussen logiese, lineêre en fisiese geheue-adresse.

    Briefly describe the relationship between logical, linear and physical memory addresses.

    > Logical addresses are generated by the CPU while physical addresses are generated by the memory unit. A linear address is an intermediary address, often generated from the logical address when using segmentation and will eventually be translated into a physical address.

    3

13. Beantwoord die volgende vrae in verband met assosiatiewe geheue (buffergeheue of TLB) wat dikwels gebruik word om die implementering van virtuele geheue te ondersteun.

    Answer the following questions regarding associative memory (translation lookaside buffer or TLB) that is often used to support the implementation of virtual memory.

    (a) Beskryf die waardes wat in die assosiatiewe geheue gestoor word.

    Describe the values that are stored in the associative memory.

    > A TLB entry consists out of a key that represents a page number and value field which holds the frame address of a page.

    2

    (b) Verduidelik kortliks die verskil tussen die werking van assosiatiewe geheue en die hoofgeheue van 'n rekenaar.
    Describe briefly the difference between the functionality of associative memory and the main memory of a computer.

    > The TLB consists out of a small amount of high speed cache memory and usually reside on the processor itself. Access to the TLB is much faster compared to main memory.

    2

    (c) Verduidelik kortliks wat gebeur met die assosiatiewe geheue se inhoud wanneer 'n nuwe proses geaktiveer word.
    Describe briefly what happens to the contents of the associative memory when a new process is activated.

    > The TLB must be flushed whenever a new process is scheduled. If not, the new process may generate addresses that are incorrectly resolved in the TLB, leading to unpredictanble behaviour or even a system crash. The TLB will gradually be filled with new entries from the addresses generated by the new process, until it is preempted and another process is scheduled.

    2

14. 'n Sekere verwerker ondersteun 'n 32-bis logiese adresruimte wat afbeeld op 'n 32-bis fisiese adresruimte. Die verwerker ondersteun dubbelvlak bladsyvertaling van logiese adresse na fisiese adresse. Die logiese adres word verdeel in 'n 10-bis gids indeks (bisse 31..22), 'n 10-bis bladsy-indeks (bisse 21..12) en 12-bis verplasing (bisse 11..0). Figuur 1 toon sekere bladsye wat tans aan proses P_0 toegeken is.

    'n Certain processor supports a 32-bit logical address space that maps on to a 32-bit physical address space. The processor supports a two level page translation mechanism that translates logical addresses to physical addresses. The logical address is divided into three components: a 10-bit directory index (bits 31..22), a 10-bit page table index (bits 21..12) and a 12-bit displacement (bits 11..0). Figure 1 shows selected pages currently allocated to process P_0.

(a) Watter voordeel word verky deurdat bogenoemde verwerker 'n dubbelvlak bladsyvertalings meganisme ondersteun?
What advantage does the two level page translation mechanism offer?

> Less memory is required to store the page tables. Instead of allocating a single large table to map all the pages, a number of smaller tables can be created as determined by the memory requirements of a process.

2

(b) Hoeveel geheue kan deur bogenoemde verwerker adresseer word? Verskaf u antwoord in megagrepe (Mb).
How much memory can be addressed by the aforementioned processor. Your answer must give the total in megabytes (Mb).

> $\frac{2^{32}}{1024 \times 1024} = 4096\text{Mb}$

1

(c) Gestel dat P_0 'n logiese adres, `02001FB8H`, genereer. Wat is fisiese adres (in heksadesimaal) waarop die logiese adres afbeeld? Toon alle stappe in die vertalingsproses asook enige berekeninge.
Suppose that P_0 generates the logical address `02001FB8H`. *What is the corresponding physical address (in hexadecimal)? Show all steps of the translation process, as well as any calculations.*

> - $02001FB8_{16} = 00000010000000000001111110111000_2$
> - Directory index $= 0000001000_2 = 8_{10}$
> - Page Table $= 0000000001_2 = 1_{10}$
> - Offset $= 111110111000_2 = FB8_{16}$
> - Entry 8 of the directory gives the base addres for the page table, `8000H`.
> - The page frame address stored at entry 1 of the page table at address `8000H` is `32000H`
> - The physical address is $32000_{16} + FB8_{16} = 32FB8_{16}$

4

(d) Wat is die grootte (in Kb) van die grootste aaneenlopende stuk fisiese geheue wat tans aan P_0 toegeken is?
What is the largest amount of contigious physical memory (in Kb) currently allocated to P_0?

> Three adjacent pages (`6A00H`, `6B00H` and `6C00H`) are allocated to P_0. Largest contigious block of memory is $3 \times 4 = 12K$.

1

(e) Watter verandering sal nodig wees ten opsigte van adres generering en vertaling indien die vervaardigers van die verwerker die bladsy grootte wil uitbrei van 4K na 16K, maar die woord grootte van adresse steeds beperk tot 32-bisse?
The designers of the processor wish to alter their addressing scheme to support 16K pages instead of 4K pages. What changes would be required to support 16K pages if the designers wish to keep the 32-bit word size for addresses?

Supporting 16K pages will require that the number of bits used for the offset be increased to $\log_2 16384 = 14$ bits. Keeping the 32-bit address scheme will require the reduction of the page table and directory components of the logical addresses. However, the page tables and page directly are themselves allocated in memory pages, but reducing the number of bits for each component will result in the low utilization of pages. For example, reducing the directory and page index to 9 bits will only allow $2^9 = 512$ entries per table, even though 16K pages can accommodate $\frac{16384}{4} = 4096$ entries. Increasing the page table field to 12 bits and decreasing the directory field to 6 bits will allow the system to still address 4GB of memory: $4096 \times 1024 \times 1024 = 16384 \times 4096 \times 64$

3

15. 'n Proses genereer die volgende reeks bladsyverwysings: 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2. Die bedryfstelsel kan net drie bladsye op enige gegewe oomblik in geheue stoor. Beantwoord nou die volgende vrae met verwysing na die voorafgenoemde proses se gedrag.

A process generates the following sequence of page references while executing: 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2. The operating system can only store 3 pages in memory at any given time. Answer the following questions with regards to the preceeding process' behaviour.

(a) Veronderstel die bedryfstelsel gebruik 'n strategie van aandringpaginering. Wat is die trefkoers indien 'n optimale bladsyvervangingsalgoritme gebruik word? Toon die inhoud van die bladsytabel na die laaste geheueverwysing.

Assume that the operating system employs a strategy of demand paging. What will the hit ratio be if an optimal page replacement algorithm is used? Show the contents of the page table after the last memory reference.

The hit ratio is $\frac{6}{12} = 0.5$.

		H			H		H	H		H	H
2	3	2	1	5	2	4	5	3	2	5	2
	2	3	2	2	5	5	4	5	3	2	5
			3	3	3	3	3	4	5	3	3

4

(b) Hoeveel bladsyfoute word gegenereer deur die proses indien 'n LRU algoritme gebruik word in samewerking met die aandringpaginering strategie? Toon die inhoud van die bladsytabel na die laaste geheueverwysing.

How many page faults are generated by the process if a basic LRU algorithm is used in conjunction with the demand paging strategy? Show the contents of the page table after the last memory reference.

A total of 7 page faults will occur.

F	F		F	F		F		F	F		
2	3	2	1	5	2	4	5	3	2	5	2
	2	3	2	1	5	2	4	5	3	2	5
			3	2	1	5	2	4	5	3	3

3

(c) Wat sal die trefkoers wees indien die stelsel 'n LRU algoritme kombineer met 'n strategie van vooruitpaginering deur die eerste drie bladsye (1, 2 en 3) in geheue te plaas voordat die proses begin uitvoer?

What will the hit ratio be if the system uses an LRU algorithm and employs a strategy of prepaging by loading the first three pages (1, 2 and 3) into memory before the process executes?

The hit ratio is $\frac{8}{12} = 0.67$.

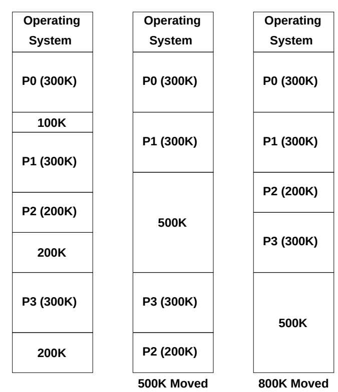
	H	H	H	H		H		H			H	H
1	2	3	2	1	5	2	4	5	3	2	5	2
2	1	2	3	2	1	5	2	4	5	3	2	5
3	3	1	1	3	2	1	5	2	4	5	3	3

2

16. Verduidelik kortliks die konsep van geheuekompaktering en bespreek hoe sodanige meganisme die effektiwiteit van 'n stelsel kan beïnvloed. Lig u verduideliking toe met 'n voorbeeld.

Briefly explaing the concept of memory compaction and discuss its influence on the effeciency of a system by way of an example.

Dynamically allocating and releasing memory leads to external fragmentation. When the system receives a request to allocate a contigious block of memory, it must determine whether or not enough free space is available. Due to fragmentation, the system may have enough free space, but not a single block large enough to accommodate the request. The system may then attempt to compact memory by relocating the allocated blocks until a single block of the desired size is obtained. However, depending on which block is relocated first, the system will either minimize or maximize the number of operations. The system may also swap a process back to secondary storage to create a large open block in memory although it may prove slow.



5

Totaal: / *Total:* 90

Bladsygids
Page Directory

0	14000H
1	3000H
2	EC000H
8	8000H
32	5000H
128	7000H

Bladsytabel (14000H)
Page Table (14000H)

0	6C000H
1	6B000H
2	70000H
3	A1000H

Bladsytabel (5000H)
Page Table (5000H)

0	2000H
1	F8000H
2	D2000H
3	12000H

Bladsytabel (7000H)
Page Table (7000H)

0	1000H
1	14000H
2	F3000H
3	24000H

Bladsytabel (EC000H)
Page Table (EC000H)

0	6A000H
1	44000H
2	4000H
3	9A000H

Bladsytabel (8000H)
Page Table (8000H)

0	87000H
1	32000H
2	6000H
3	C4000H

Bladsytabel (3000H)
Page Table (3000H)

0	100000H
1	21000H
2	C9000H
3	FF000H

Figure 1: Inhoud van geselekteerde bladsytabelle van proses P_0. *Contents of selected page tables of process P_0.*