



29.03.2010

Prof. Dr. Frank Bellosa
Philipp Kupferschmied

Aufgabe 1: Zum Aufwärmen/Question 1: Warmup

- a) Geben Sie das Ergebnis der folgenden Rechenoperation in hexadezimaler Darstellung an. („<<“ ist eine Bitshift-Operation, „|“ ein bitweises „oder“.)

2 pt

*What is the result of the following arithmetic operation in hexadecimal notation?
(<< is a bitshift operator, | is a bit-wise “or”.)*

$(0x00FC \ll 4) \mid 0x003F$

Lösung: $0xFFFF$ (2 P) .

Korrekte „<<“ oder „|“-Operation, aber falsches Ergebnis: (1 P) .

- b) Welche der folgenden Aussagen sind korrekt, welche sind inkorrekt?

5 pt

Which of the following statements are correct, which are incorrect?

korrekt/
correct

inkorrekt/
incorrect

☐☒

trap ist eine privilegierte Instruktion.
trap is a privileged instruction.

☒☐

Auf Einprozessorsystemen ist zu jedem Zeitpunkt höchstens ein Prozess im Zustand „rechnend“ („running“).
On a uniprocessor system, at most one process is in the state “running” at any given time.

☐☒

Paging verhindert internen Verschnitt innerhalb einer Seitenkachel.
Paging prevents internal fragmentation within a page frame.

☒☐

Durch Verwendung von ASIDs entfällt die Notwendigkeit, den TLB beim Adressraumwechsel vollständig invalidieren zu müssen.
The use of ASIDs eliminates the necessity to completely flush the TLB when switching address spaces.

☐☒

RAID 0 mit mindestens 2 Platten erlaubt nach dem Totalausfall einer Platte eine vollständige Wiederherstellung aller Daten.
Using RAID 0 with at least 2 disks facilitates recovery of all data if one disk fails completely.

Ausschließlich die korrekte Antwort markiert (1 P) , sonst (0 P) (pro Aussage).

- c) Erklären Sie die Begriffe „Systemaufruf“, „Exception“ und „Interrupt“. Gehen Sie insbesondere darauf ein, wann und von wem das jeweilige Ereignis ausgelöst wird.

3 pt

Explain the terms “system call”, “exception”, and “interrupt”. Point out when and by whom such an event is caused.

Lösung:

Systemaufruf: Wird von einem Userlevel-Programm getätigt **(0.5 P)**, um vom Kernel bereitgestellte Dienste aufzurufen **(0.5 P)** (z.B. Öffnen einer Datei, Erstellen eines neuen Prozesses, ...)

Exception: Von der Hardware ausgelöst **(0.5 P)**, falls bei der Ausführung einer Instruktion ein Fehler auftrat **(0.5 P)** (z.B. Division durch 0, ungültiger Speicherzugriff, ...)

Interrupt: Ausgelöst von Hardwarekomponenten **(0.5 P)**, die damit ein bestimmtes Ereignis signalisieren **(0.5 P)** (z.B. anliegende Daten, Ende eines Zeitintervalls, ...)

*Je **(0.5 P)** für eine korrekte Antwort auf „wann“ und „von wem“; für letzteres genügt jeweils ein Beispiel.*

- d) Erläutern Sie die beiden Hauptprobleme, die mit virtuell indizierten, virtuell getagten Caches auftreten können.

2 pt

Explain the two main problems that can occur with virtually indexed, virtually tagged caches.

Lösung:

Mehrdeutigkeiten (ambiguity): Eine virtuelle Adresse wird zu verschiedenen Zeitpunkten oder in verschiedenen Adressräumen **(0.5 P)** auf unterschiedliche physische Adressen abgebildet. Ohne Invalidierung der entsprechenden Cachezeilen könnte der Cache „veraltete“ Daten enthalten **(0.5 P)**.

Bedeutungsgleichheit (alias): Verschiedene virtuelle Adressen werden auf die gleiche physische Adresse abgebildet. Dadurch kann es zu Inkonsistenzen kommen **(1 P)**.

*Je **(0.5 P)** für die Begriffe „Mehrdeutigkeiten/ambiguity“ und „Bedeutungsgleichheit/alias“, sofern sie nicht erläutert werden.*

**Total:
12 pt**

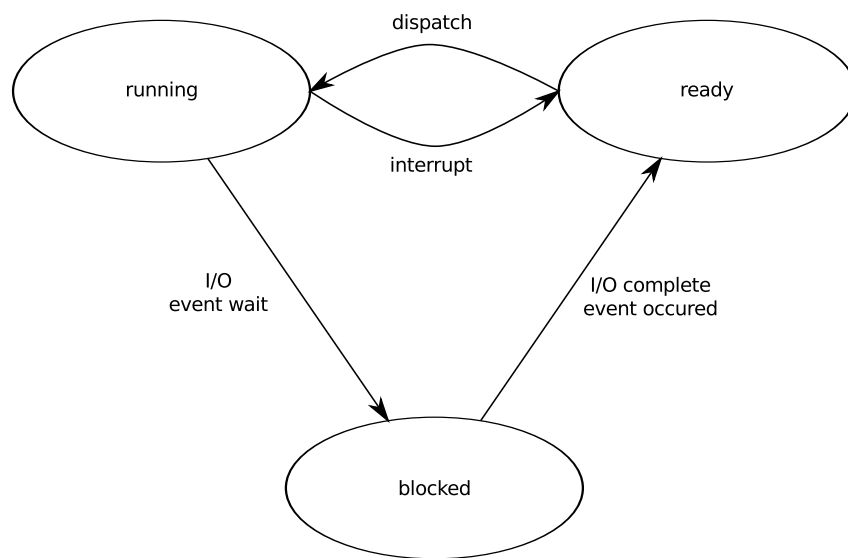
Aufgabe 2: Prozesse/Question 2: Processes

- a) Nehmen Sie an, dass ein Betriebssystem die drei Taskzustände „rechnend“ (“running”), „bereit“ (“ready”) und „blockiert“ (“blocked”) unterstützt. Stellen Sie grafisch dar, zwischen welchen Zuständen Übergänge möglich sind und geben Sie jeweils an, von welchen Ereignissen diese Übergänge ausgelöst werden.

2 pt

Consider an operating system that supports the three task states “running”, “ready”, and “blocked”. Depict the possible state transitions and the events that cause them.

Lösung:



Je **(0.5 P)** pro korrekt eingezeichnetem und korrekt beschriftetem Zustandsübergang. „Scheduling“, „short time scheduling“ o.ä. sind beim Übergang von „rechnend“ nach „bereit“ kein auslösendes Ereignis **(0 P)**.

Die alleinige Angabe von „I/O“ oder „I/O complete“ beim Übergang von „rechnend“ nach „blockierend“ bzw. von „blockierend“ nach „bereit“ reicht nicht aus, da auch andere Ereignisse diese Übergänge auslösen können **(0 P)**.

- b) Angenommen, Sie sollten einen mehrfädigen (multithreaded) Webserver entwickeln, bei dem ein Thread Anfragen von Clients entgegennimmt und diese unter einer Menge von Arbeiter-Threads verteilt. Jeder Arbeiter-Thread bearbeitet seine Anfrage, indem er die benötigten Daten mittels eines blockierenden Systemaufrufs von der Festplatte liest und dann an den Client zurückschickt. Welches der drei aus der Vorlesung bekannten Threading-Modelle würden Sie in diesem Szenario bevorzugen? Begründen Sie Ihre Antwort.

2 pt

Assume you have to write a multithreaded web server which uses one of its threads to receive all incoming client requests and to dispatch them among a set of worker threads. Each worker thread processes a request by issuing a blocking system call to read the required data from the hard disk and then sending the data back to the

client. Which of the three threading models that were introduced in the lecture would you prefer in this scenario? Explain your answer.

Lösung:

One-to-One (1 P) . Bei Many-to-One würde ein blockierender Systemaufruf eines Threads die gesamte Anwendung blockieren (1 P) . Many-to-Many ist auch denkbar, aber eigentlich nur sinnvoll, wenn der Kernel die Userlevel-Threads über Blockierungen benachrichtigen kann (etwa via Scheduler Activations).

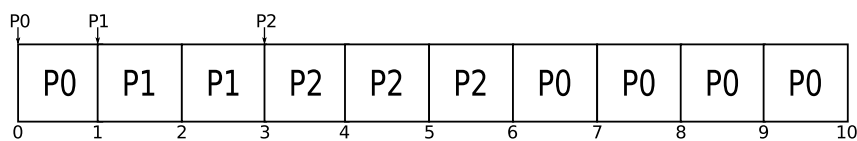
Prozess/process	Burstlänge/burst length	Ankunftszeit/arrival time
P_0	5	0
P_1	2	1
P_2	3	3

- c) Betrachten Sie die obige Tabelle. Zeichnen Sie ein Gantt-Diagramm für den Fall dass präemptives SJF als Scheduling-Strategie verwendet wird.

3 pt

Consider the table above. Draw a Gantt chart of a preemptive SJF schedule.

Lösung:



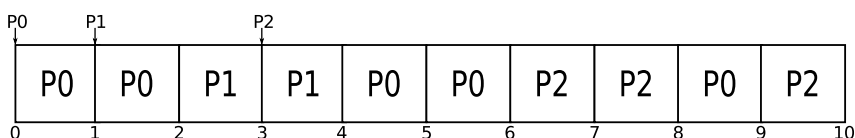
Bewertung: s. nächste Teilaufgabe

- d) Nun soll präemptives Round-Robin-Scheduling mit einer Zeitscheibenlänge von 2 zur Einplanung der Prozesse aus obiger Tabelle verwendet werden. Neu hinzugekommene Prozesse sollen an das Ende der Bereitwarteschlange (ready queue) angehängt werden. Zeichnen Sie auch hierfür ein Gantt-Diagramm.

3 pt

Now let the processes from the table above be scheduled using preemptive round-robin scheduling and a timeslice length of 2. New processes are added to the tail of the ready queue. Again, draw the corresponding Gantt chart.

Lösung:



Bis zu (-0.5 P) , wenn bei Teilaufgabe (c) oder (d) keine Zeitskala angegeben ist.

Der häufige Fehler, in Teilaufgabe (d) zum Zeitpunkt 4 P_2 anstatt P_0 einzuplanen, führt zu (-1.5 P) .

- e) Zählen Sie zwei Vorteile von Lottery Scheduling gegenüber prioritätsbasierten Ansätzen auf.

2 pt

List two advantages of lottery scheduling over priority-based approaches.

Lösung:

- Verhungern wird verhindert: Solange jeder Prozess mindestens ein Ticket hat, kommt er irgendwann zur Ausführung **(1 P)**.
- Ticketweitergabe erlaubt eine genauere Abrechnung von Rechenzeit: Ein Clientprozess kann Tickets an einen Serverprozess weitergeben, sodass dieser Rechenzeit des Clients (statt eigener Rechenzeit) „verbraucht“ **(1 P)**.
- Beim Lottery Scheduling wirken sich Lastveränderungen (Hinzunahme oder Entfernen von Prozessen) gleichmäßig und proportional zum jeweiligen Ticketbesitz der noch laufenden Prozesse aus (graceful degradation) **(1 P)**.

**Total:
12 pt**

Aufgabe 3: Synchronisation/Question 3: Synchronization

- a) Implementieren Sie die `wait()`-Operation eines blockierenden Semaphors in Pseudocode. Die Funktion `block()` kann benutzt werden, um den aktuellen Prozess zu blockieren und den Scheduler aufzurufen. Die globale Variable `cur_process` bezeichnet den aktuell laufenden Prozess. Der Datentyp `List` stellt die Operationen `append(elem)` und `remove()` zur Verfügung, die dazu verwendet werden können, ein Element an die Liste anzuhängen bzw. das erste von dieser zu entfernen und zurückzugeben.

Sie brauchen **nicht** sicherzustellen, dass der Code atomar ausgeführt wird!

4 pt

Implement the `wait()` operation of a blocking semaphore in pseudo code. The function `block()` can be used to block the currently running process and to activate the scheduler. The global variable `cur_process` identifies the currently running process. The data type `List` offers the operations `append(elem)` and `remove()` that can be used to append an element to the list and to remove and return the first element from the list, respectively.

*You do **not** have to make sure that the code is executed atomically!*

```

class Semaphore {
    int val;
    List waitlist;

    void wait() {
        this.val--;
        if ( this.val < 0 ) {
            this.waitlist.append(cur_process);
            block();
        }
    }

    void wait_alternative() {
        if ( this.val <= 0 ) {
            this.waitlist.append(cur_process);
            block();
        } else {
            this.val--;
        }
    }
}

```

- Je bis zu **(2 P)** für korrekte Behandlung der Fälle „Semaphore frei“ bzw. „belegt“.
– Ausnahme: nur `this.waitlist.append(); block();`: **(0 P)** .
- Je **(-1 P)** wenn `this.val--` oder `this.waitlist.append()` erst nach `block()`.
- Verwaltung der `waitlist` grob falsch: **(-1 P)** .
- Falsche Relation im Vergleich: **(-0.5 P)** .

b) Nennen und erklären Sie die vier notwendigen Bedingungen für Deadlocks.

2 pt

List and explain the four necessary conditions for deadlocks.

Lösung:

Mutual exclusion (aka exclusiveness): Eine Ressource kann nicht gleichzeitig von mehreren Prozessen benutzt werden **(0.5 P)** .

Hold and wait: Ein Prozess, der bereits mindestens eine Ressource hält, wartet auf mindestens eine andere Ressource **(0.5 P)** .

No preemption: Zugeteilte Ressourcen können einem Prozess nicht wieder entzogen werden, er muss diese selbst freigeben **(0.5 P)** .

Circular wait: Es gibt eine Menge von Prozessen $\{P_0, P_1, \dots, P_n\}$, wobei P_0 auf eine Ressource wartet, die P_1 hält, P_1 auf eine Ressource wartet, die P_2 hält usw. und P_n auf eine Ressource wartet, die P_0 hält **(0.5 P)** .

Aufzählung der vier Punkte ohne Erklärung: **(0.5 P)** .

- c) Betrachten Sie ein System mit drei Arten von Ressourcen (R_1 , R_2 und R_3) und drei Prozessen (P_1 , P_2 , und P_3). Analysieren Sie den untenstehenden Schnappschuss des Systems. Befindet sich das System in einem sicheren Zustand? Begründen Sie Ihre Antwort.

3 pt

Consider a system with three resource types (R_1 , R_2 , and R_3), and three processes (P_1 , P_2 , and P_3). Analyze the system snapshot below. Is the system in a safe state? Why, or why not?

Allocation:		R_1	R_2	R_3	Max:		R_1	R_2	R_3	Available:	R_1	R_2	R_3
	P_1	1	2	0		P_1	5	2	0		2	1	1
	P_2	1	0	1		P_2	1	3	3				
	P_3	1	1	1		P_3	2	2	2				

Lösung:

Work = Available = (2, 1, 1)

Need = Max - Allocation =		R_1	R_2	R_3
	P_1	4	0	0
	P_2	0	3	2
	P_3	1	1	1

$Need_{P_3} \leq Work \Rightarrow P_3$ kann zu Ende laufen **(1 P)**.

Danach ist Work = (3, 2, 2), somit $Need_{P_1} \not\leq Work$ und $Need_{P_2} \not\leq Work \Rightarrow$ weder P_1 noch P_2 kann zu Ende laufen **(1 P)**, der Zustand ist also nicht sicher **(1 P)**.

- d) Führt ein unsicherer Zustand immer zu einem Deadlock? Warum, oder warum nicht?

1 pt

Does an unsafe state always lead to a deadlock? Why, or why not?

Lösung: Nein **(0.5 P)**, da ein Prozess nicht notwendigerweise das Maximum an Ressourcen anfordert / ggf. eine abwechselnde Ausführung erfolgreich ist **(0.5 P)**.

- e) Diskutieren Sie die Vor- und Nachteile von Deadlock-Vermeidung (avoidance) gegenüber Deadlock-Erkennung (detection).

2 pt

Discuss advantages and disadvantages of deadlock avoidance compared to deadlock detection.

Lösung:

Deadlock avoidance:

- + Es wird sichergestellt, dass keine Deadlocks auftreten **(0.5 P)**.
- + Deadlocks müssen nicht behoben werden **(0.5 P)**.
- Überprüfung bei jeder Ressourcenzuteilung nötig **(0.5 P)**.
- Maximalbedarf an Ressourcen pro Prozess muss vorab bekannt sein **(0.5 P)**.
- Ggf. unnötige Verzögerung von Ressourcenanforderungen **(0.5 P)**.

**Total:
12 pt**

Aufgabe 4: Speicher/Question 4: Memory

Betrachten Sie ein System, welches virtuelle Adressen mittels zweistufiger Seitentabellen auf physische Adressen umsetzt. Sowohl virtuelle als auch physische Adressen sind 32 Bit breit. Jede Seitentabelle besteht aus 1024 Einträgen zu jeweils 4 Byte. Die Seitengröße beträgt 4096 Bytes. Der Speicher ist byte-adressierbar.

Consider a system that uses two-level page-tables to map virtual addresses to physical addresses. Both virtual and physical addresses are 32 bit wide. Each page-table consists of 1024 entries of 4 bytes. The page size is 4096 bytes. The memory is byte addressable.

- a) In welche Teile muss eine virtuelle Adresse zerlegt werden, um eine Adressübersetzung durchführen zu können? Aus wie vielen Bits besteht jeder dieser Teile unter den oben genannten Voraussetzungen?

3 pt

Into which parts must a virtual address be split so that it can be translated into a physical address? How many bits are required for each part under the assumptions stated above?

Lösung:

Je (0.5 P) für korrekte Feldbreiten (10/10/12 Bit).

Je (0.5 P) für korrekte Feldbezeichnungen:

- {page directory, 1st-level, top, outer} index / table number
- {page table, 2nd-level, inner} index / page number
- offset

Falsche Reihenfolge: (-0.5 P)

Nur Seitennummer (22 Bit) und Offset (10 Bit) (1 P), falsche Feldgrößen (0.5 P).

Kein Hinweis, dass die 10 Bit breiten Felder Indizes sind (-0.5 P).

- b) Wozu dient ein TLB?

1 pt

What is the purpose of a TLB?

Lösung: Ein TLB beschleunigt die Übersetzung von virtuellen in physische Adressen (1 P).

- c) Bestimmen Sie die effektive Speicherzugriffszeit unter der Annahme, dass ein TLB-Lookup 10 ns und ein Speicherzugriff 200 ns dauert und dass der TLB eine durchschnittliche Trefferrate von 50 % hat. Nehmen Sie an, dass das System keine weiteren Caches außer dem TLB besitzt.

3 pt

Calculate the effective memory access time, assuming that a TLB lookup takes 10 ns, a memory access takes 200 ns, and that the TLB has an average hit ratio of 50%. Assume that the system uses no caches apart from the TLB.

Lösung:

Grundformel: $0.5 * \text{Hit} + 0.5 * \text{Miss}$ (0.5 P)

Hit: $10 \text{ ns} + 200 \text{ ns} = 210 \text{ ns}$ (1 P)

Miss: $10 \text{ ns} + 200 \text{ ns} + 200 \text{ ns} + 200 \text{ ns} = 610 \text{ ns}$ (1 P)

Richtiges Ergebnis (ggf. als Folgefehler): 410 ns (0.5 P)

Je (-0.5 P) für fehlenden Zugriff auf TLB, zweite Tabellenstufe oder Hauptspeicher.

- d) Gegeben sei die Seitentabellenhierarchie aus Abbildung 1. Zerlegen Sie die virtuelle Adresse **0x4007E800** gemäß Aufgabe 4 a). Berechnen Sie dann die physische Adresse, auf die die virtuelle Adresse abgebildet wird.

5 pt

Consider the page-table hierarchy depicted in Figure 1. Decompose the virtual address **0x4007E800** into its parts according to question 4 a). Calculate the physical address to which it is mapped.

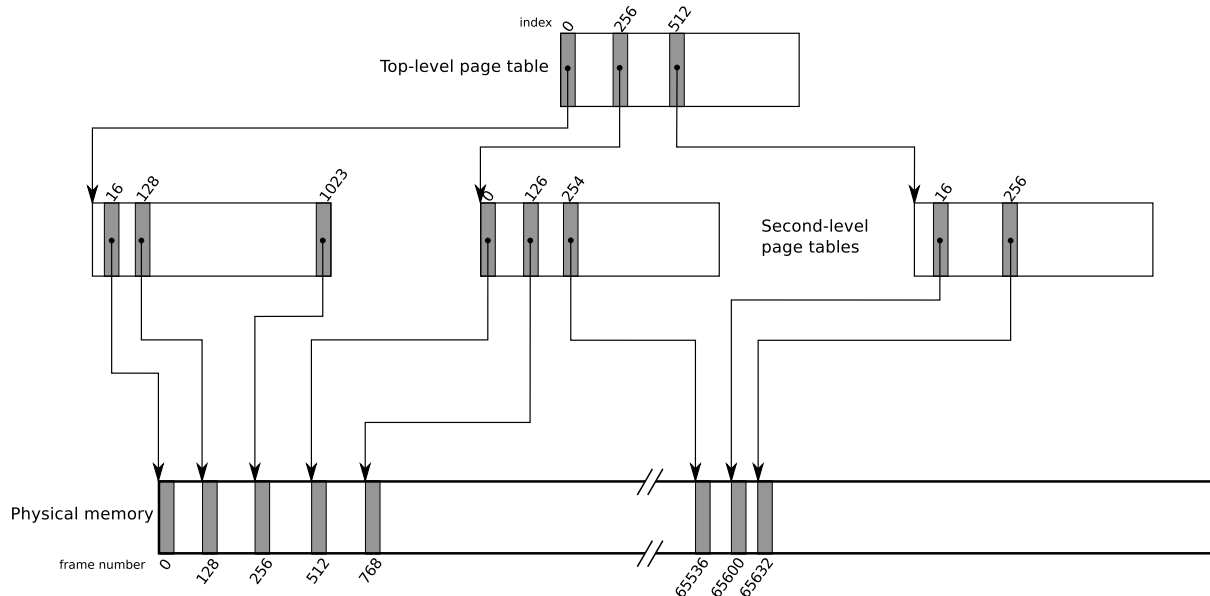


Abbildung 1: Seitentabellenhierarchie/Page-table hierarchy

Lösung:

$0x4007E800_{16} = 0100\ 0000\ 00\ |\ 00\ 0111\ 1110\ |\ 1000\ 0000\ 0000$

31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0

1st-level index = $0x100 = 256$	2nd-level index = $0x7E = 126$	page offset = $0x800$
------------------------------------	-----------------------------------	-----------------------

Je **(1 P)** für korrekte Indizes, **(0.5 P)** falls um 1 Bit daneben.

Korrektes Offset **(0.5 P)**.

Falls Zerlegung falsch, aber „korrekt“ gemäß Teilaufgabe (a) **(0.5 P)**.

Aus der Abbildung kann man dann entnehmen, dass die gesuchte virtuelle Adresse auf die physische Kachel 768 abgebildet wird **(0.5 P)**.

Gefragt war allerdings die physische Adresse, die sich aus Kachelnummer * Kachelgröße + Offset ergibt, also:

Kachelnummer = $768 = 512 + 256 = 0011\ 0000\ 0000 = 0x300$

Kachelgröße = $4096 = 0x1000 = 2^{12}$

Kachelnummer * Kachelgröße = $0x300 \ll 12 = 0x300000$

Physische Adresse = $0x300000 + 0x800 = 0x300800$ **(2 P)**

Falls 768 falsch umgerechnet (und Fehler einigermaßen ersichtlich), aber sonst richtig **(1.5 P)**. Falls nicht explizit ausgerechnet **(1 P)**. Formulierung „Offset 2048 in Frame 768“: **(0.5 P)**. Framenummer + Offset: **(0 P)**.

Total:
12 pt

Aufgabe 5: Dateisysteme/Question 5: File Systems

- a) Welche Zugriffsrechte haben Besitzer, Gruppe und andere an der Datei `exam.tex` auf einem Unix-Dateisystem, nachdem untenstehendes Kommando erfolgreich ausgeführt wurde?

2 pt

State the respective access rights of owner, group, and others to the file `exam.tex` on a Unix file-system after the command below has been successfully executed.

```
chmod 640 exam.tex
```

Lösung:

Besitzer: `r w` – (lesen und schreiben) **(1 P)**
Gruppe: `r` – – (lesen) **(0.5 P)**
Andere: – – – (keine Rechte) **(0.5 P)**

Je **(-0.5 P)** pro inkorrekt angegebenem Recht.

- b) Welche Einträge müssen die Metadaten einer Datei mindestens enthalten, wenn „contiguous allocation“ zur Reservierung von Plattenblöcken verwendet wird?

1 pt

Which entries must at least be present in the metadata of a file if “contiguous allocation” is used to allocate disk blocks?

Lösung: Anfangsadresse **(0.5 P)** und Größe **(0.5 P)**.

Je **(-0.5 P)** pro falscher/überflüssiger Angabe.

Letzter Block/Anzahl an Blöcken statt Größe ist falsch, da Dateigrößen kein exaktes Vielfaches der Blockgröße sein müssen.

- c) Was ist ein „einfach indirekter Block“ und wozu wird er benötigt?

1 pt

What is a “single indirect block”, and what is its purpose?

Lösung: Ein „einfach indirekter Block“ ist ein Plattenblock, der Zeiger auf (bzw. Adressen von) weiteren Plattenblöcken enthält **(0.5 P)**. Er wird verwendet, um Dateien verwalten zu können, die mehr Blöcke belegen, als eine Inode direkt referenzieren kann **(0.5 P)**.

- d) Gegeben sei der untenstehende Ausschnitt aus einer FAT. Welche Blöcke werden von der Datei `f` belegt, die mit Block 3 beginnt? Geben Sie die Blöcke in der korrekten Reihenfolge an. (–1 markiert das Ende einer Datei; die erste Zeile der Tabelle enthält die Nummer der entsprechenden Tabellenspalte.)

1 pt

Consider the cutout of a FAT given below. Which blocks are occupied by file `f` that starts with block 3? List the blocks in the correct order. (–1 indicates the end of a file; the first line of the table contains the number of the respective column.)

0	1	2	3	4	5	6	7
6	7	4	1	0	2	-1	5

Lösung: 3, 1, 7, 5, 2, 4, 0, 6 **(1 P)**

Je **(-0.5 P)** pro vergessenem Block (auch Block 3) oder Fehler in der Reihenfolge.

- e) Angenommen Sie erstellen einen symbolischen Link s zu einer existierenden Datei g . Wie ändert sich dadurch der Linkzähler in der Inode von g ? Was passiert mit s , wenn g gelöscht wird?

2 pt

Assume you create a symbolic link s to an existing file g . How does this affect the link count in the inode of g ? What happens to s when g is deleted?

Lösung: Der Linkzähler ändert sich durch symbolische Links nicht (1 P) .

Nach dem Löschen von g zeigt der Link ins Leere, bleibt aber unverändert bestehen (1 P) . (Für den zweiten Teil der Frage genügt eine der beiden Angaben.)

- f) Gegeben seien Plattenanfragen für die Zylinder 1, 17, 23, 40, 42, 50, 70, 105. Der Schreib-/Lesekopf befindet sich momentan auf Zylinder 45. Geben Sie an, in welcher Reihenfolge die einzelnen Zylinder angesteuert werden, wenn als Plattenschedulingstrategie SSTF verwendet wird.

2 pt

Given a set of disk requests for the cylinders 1, 17, 23, 40, 42, 50, 70, 105. The disk head is currently on cylinder 45. In what order are the requests served if SSTF is used for disk scheduling?

Lösung: (45), 42, 40, 50, 70, 105, 23, 17, 1 (2 P)

Je (-0.5 P) pro Fehler in der Reihenfolge.

- g) Was ist das Hauptproblem, das bei Verwendung von SSTF auftreten kann?

1 pt

What is the main problem that can occur with SSTF?

Lösung: Anfragen nach Zylindern „am Rand“ können verhungern, falls ständig Anfragen nach Zylindern „in der Mitte“ vorliegen (1 P) .

- h) Erklären Sie die RAID-Level RAID1 und RAID5.

2 pt

Explain the RAID levels RAID1 and RAID5.

Lösung:

RAID1: Mirroring. Daten werden auf mehrere Platten gespiegelt (1 P) .

RAID5: Block-Level-Parität (0.5 P) mit verteilten Paritätsblöcken (0.5 P) .

Total:
12 pt