Was versteht man unter einem Microkernel? Welche Services stellt ein Microkernel zur Verfügung? Welche Vor- bzw. Nachteile ergeben sich bei der Verwendung eines Microkernel Betriebssystems? (5)

Microkernel stellen eine Betriebssystem Architekturart dar, bei der nur die wichtigsten Funktionen im Kernel enthalten sind, alles andere wird auf User-Level ausgeführt und kann mit Kernel kommunizieren

Services:

- Process Switching
- Basic Memory Management
- Interrupts und Hardware Access (I/O)
- Nachrichtenaustausch und –kontrolle

Vorteile:

- Einheitliches Interface
- Flexibilität und Erweiterbarkeit
- Portabilität
- Unterstützung von Verteilung

Nachteile:

- Performance eher schlecht

Was versteht man unter einem Process Control Block ? Beschreiben Sie, aus welchen Teilen der PCB besteht und welche Informationen in diesen Teilen jeweils verwaltet werden. (6)

- Process Identification: Eindeutige Prozessnummer, Benutzerkennung, Nummer des Prozesses, der den Prozess generiert hat (Parent Process Identifier)
- Processor State Information: Registerinhalte, Kontroll- und Statusregister (Befehlszähler, Program Status Word (PSW) [Kontroll-, Modusinformation, Status-Bits…]), Stack Pointers
- Process Control Information: Scheduling und Zustandsinformation (Zustand in dem sich der Prozess befindet, Priorität und Schedulinginformation, Ereignis, auf das der Prozess wartet), Querverweise auf andere Prozesse (Realisierung von Prozess-Queues, Verweis auf Parent, Child,…), Interprozesskommunikation IPC (Flags, Signale, Verweise auf Nachrichten), Privileges, Memory Management (Verweise auf Segment- oder Seitentabellen), Ressourcen (verwendete [geöffnete Files, I/O Devices], bisher konsumierte [CPU Zeit, I/O, etc.]

Worin liegt der grundlegende Unterschied zwischen Prozessen und Threads? Welcher Vorteil ergibt sich aus der Einführung von Threads für den Benutzer und worauf muss der Benutzer achten? (4)

Entkopplung von Ressourcenverwaltung und Dispatching (kurzfristiges Scheduling) →

- Process (Task): Einheit der Ressourcenverwaltung
    - Virtueller Adressraum mit Process Image
    - Speicherschutz, Files I/O Ressourcen
- Thread (Lightweight Process): Einheit für das Dispatching
    - Ausführungszustand (Running, Ready, …)
    - Kontext (wenn nicht gerade laufend)
    - Stack
    - Thread-lokale statische und lokale Variable
    - Zugriff auf Prozessspeicher und Ressourcen

- Vorteile Threads:
    - Thread Erzeugung benötigt weniger Zeit
    - Umschaltung zwischen Threads geht schneller als ein Process Switch
    - Kommunikation zwischen Threads eines Prozesses ohne Einschaltung des Kernels, aber: Synchronisation notwendig!!!!
- Einsatzbereiche: Applikationen, die zusammengehörige Menge von Abarbeitungseinheiten bilden
    - File Server LAN
        - Mehrere Requests in kurzer Folge
        - Ein Thread für jeden Request
    - Spredsheet-Programm
        - Ein Thread zeigt Menüs an und liest Inputs
        - Ein Thread führt Berechnungen und Updates aus
- User Level Threads ULT
    - Für Kernel unsichtbar
    - Applikationsspezifisches Scheduling
    - Keine Verteilung auf mehrere Prozessoren
- Kernel Level Threads KLT
    - Vom Kernel verwaltet
    - Durch mode switches langsamer als ULT
    - Von mehreren Prozessoren verwaltbar

➔ Benutzer muss bei ULTs aufpassen, dass Synchronisation in seinen Händen liegt

Was versteht man unter Deadlock Avoidance? Geben Sie zwei Strategien für Deadlock Avoidance an und beschreiben Sie diese. (4)

Es wird versucht Deadlocks gar nicht erst auftreten zu lassen, womit keine Prozesse zurückgesetzt werden müssten.

1. Process Initiation Denial: Prozess wird gar nicht erst gestartet, falls seine Resourcenanforderungen einen Deadlock hervorrufen könnten, Ressourcen müssen dadurch jedoch schon zu Beginn bekannt sein → schränkt Parallelität ein
2. Resource Allocation Denial: Ressourcenanforderung wird verweigert, falls Deadlock entstehen könnte. Über Matritzen die Ressourcen darstellen wird ein Weg gesucht, in dem alle Prozesse in einen sog. Safe state übergehen können

Was versteht man unter einem Monitor zur Prozesssynchronisation? Nennen Sie die wichtigsten Komponenten und Eigenschaften des Monitors. (4)

Ein Monitor ist ein Softwaremodul bestehend aus Prozeduren, lokalen Daten und Initialisierungsode. Prozeduren regeln Zugriff durch warten auf bestimmte Bedingungen (z.B. notfull oder notempty) → Prozesse stellen sich in Queue an, pro Porzedur kann nur ein Prozess zugriefen (java synchronized)

Eigenschaften:

- Zugriff auf lokale Variable: Monitorprozeduren
- Eintritt von Prozessen in den Monitor über Monitorprozeduren
- Max. 1 Prozess zu jedem Zeitpunk im Monitor
- Sorgt für Mutual Exclusion, kein explizites Programmieren
- Gemeinsamer Speicher ist im Monitorbereich anzulegen
- Bedingungssynchronisation über Monitorvariable
- Condition Variables: lokal, nur im Monitor zugreifbar

- cwait(c) blockiert aufrufenden Prozess bis c den Wert true annimmt
- csignal(c) setzt Prozess der auf c wartet fort → wenn Prozess wartet, führe diesen aus, falls keiner wartet, verwerfe c → wird nicht gespeichert

Was versteht man unter dem Begriff Relocation? Wofür ist Relocation von Bedeutung? (3)

Da Prozesse aufgrund von Swapping auch auf die Disk ausgelagert werden können, wäre es sehr umständlich, wenn man immer denselben Speicherbereich belegen wollen würde im Hauptspeicher → Relocation notwendig → Umwandlung von Speicheradressen im Code in tasächliche Speicheradressen!

Logical address → Nur logischer Zeiger auf Daten, muss erst auf physikalische Adresse gemappt werden

Relative address→ Adresse von einem bekannten Punkt aus, nur durch z.B. Pfad angegeben

Physical address → tastächliche Adresse im Hauptspeicher

Beschreiben Sie, wozu und wie eine Page Table verwendet wird. Geben Sie weiters an, welche Informationen in den Tabelleneinträgen einer Page Table gespeichert werden. (4)

Da nicht immer der gesamte Prozess in den Hauptspeicher passt, wird der Hauptspeicher in Frames und der Prozess in Pages unterteilt. Der Page Table enthält die zugehörige Frame Nummer für jeden Speicherzugriff des Prozesses. Dabei beinhaltet jede Speicheradresse eine PageNummer und einen Offset. Im Pagetable sind dann an der Stelle der Pagenummer die zugehörige Framenummer gespeichert und durch aneinanderhängen kann die richtige Adresse im Hauptspeicher gefunden werden. Falls nicht im Hauptspeicher → Page Fault → muss nachgeladen werden → Ersetzungsstrategie notwendig.
Zusätzliche bits ob Veränderungen vorgenommen wurden, ob im Hauptspeicher enthalten und für spezielle Infos (z.B. read oder write)

TLB (Translation Lookaside Buffer) → einige Einträge jedes verwendeten PageTables im Hauptspeicher → schnellers auffinden häufig genutzter Einträge im Page Table

Beschreiben Sie das Ziel von Disk Scheduling. Nennen Sie drei „intelligente" Disk-Scheduling Algorithmen und beschreiben Sie diese kurz. (5)

Disk Scheduling soll dabei helfen Anfragen auf die Disk so abzuarbeiten, dass die seek time möglichst kurz wird. Simple algorithmen wären FIFO, LIFO oder einfach eine zufällige Auswahl. Intelligente hingegen:

- Shortest Service Time First (SSTF): Die Anfrage mit geringster vorraussichtlicher seek time vom derzeitigen Punkt aus wird gewählt, bei gleicher einfach random
- Scan: LIFO, random und SSTF können gewisse Anfragen verhungern lassen (immer ein anderer wird ausgewählt), um dem entgegenzuwirken wandert SCAN über die Platte nur in einer Richtung und behandelt die requests der Reihe nach an passender Position, am Ende wird Richtung umgekehrt und wieder alle die passend sind werden behandelt → keine Starvation
- C-Scan: Nur mehr in eine Richtung, wenn letzter Track erreicht wird von anderer Seite wieder begonnnen → sonst werden Anfragen in der Mitte
- N-step-Scan: Nur N aufeinanderfolgende Anfragen werden mit Scan abgearbeitet → falls ein bestimmter Bereich sehr stark angefordert wird, könnte es sonst sein, dass Arm dort „stecken bleibt"
- FSCAN: verwendet zwei N-Step Queues → schneller als N-Step

Was versteht man unter Buffering? Welche Vorteile bietet es, wo liegen seine Grenzen und worauf hat man bei der Verwendung von Puffern bei der Betriebssystemimplementierung zu achten? (5)

Buffering hilft beim lesen und schreiben von I/O gebunden Daten. Das tatsächliche Lesen wird aus dem Adressraum des Prozesses ausgelagert, in den sogenannten Buffer und erst wenn dieser voll ist verschoben. Dadurch kann während der Prozess die Daten verarbeitet schon der nächste Block eingelesen werden und es kann zu keinem Deadlock führen falls der Prozess ausgelagert wird (Swapping). Falls keine weiteren Daten gelesen/geschrieben werden müssen, muss der Prozess auch nicht auf Abschluss warten, sondern kann ausgelagert werden.

Was versteht man unter einer File Allocation Table? Wie ist diese organisiert? (2)

Ein File Allocation Tabel (FAT) speichert die Aufteilung der Files innerhalb der Disk. Dazu wir der Filename, der Anfangspunkt und die Länge gespeichert. Egal ob contiguous oder chained allocation stattfindet werden nur diese Einträge benötigt → contiguous einfach Blöcke der Reihe nach durchgehen bis länge erreicht → chained solange next point folgen bis länge erreicht

Nennen Sie die drei Kategorien von Security Threats und beschreiben Sie diese. Geben Sie für jede Kategorie an, welches grundlegende Security-Ziel dadurch bedroht wird. (4)

- Denial of Service (Interruption): Availability wird bedroht → kein Zugriff auf die Daten möglich
- Exposure/Interception: Confidentiality → vertrauliche Daten werden ausspioniert → z.B. Passwörter
- Modification/Fabrication: Integrity → Daten sollen Sollzustand entsprechen und nicht gelöscht/verändert werden

Was beschreibt das Modell von Bell und LaPadula? Geben Sie die vom Modell geforderten Eigenschaften an. (4)

Beschreibt ein Modell für Regeln für den Infomationsfluss → geordnete Security Classifications (SC) für Subjects (S) und Objects (O) → top secret, secret, public → read-only, append (without reading), execute (without reading or writing), read-write

read → SC(S) >= SC(O)
append → SC(S) <= SC(O)
read-write → SC(S) = SC(O)

Erklären Sie die Aktionen, die vom Betriebssystem bei einem Process Switch durchzuführen sind. Welche Arten von Ereignissen können zu einem Process Switch führen? (3)

Umschalten des aktiven Prozesses, PCB muss gespeichert werden, Register und Stackeinträge gespeichert → Prozess soll falls möglich an genau derselben Stelle wieder fortsetzen können.

Immer wenn BS im Besitz der CPU ist möglich:

- Supervisor Call: expliziter Aufruf (z.B. I/O)
- Trap: bedingt durch aktuelle Instruktion (z.B. Auftreten eines Fehlers)
- Interrupt: Ursache außerhalb des Prozesses, Kontrolle an Interrupt Hanlder und BS

Beschreiben Sie die folgenden Strategien für das CPU Scheduling und vergleichen Sie deren Eigenschaften: Round Robin, Virtual Round Robin, Highest Response Ratio Next und Feedback Scheduling. (6)

- **Round Robin:** gleiche große Zeitscheiben zyklisch an Prozesse vergeben
- **Virtual Robin:** Auxilary Queue für I/O lastige Prozesse → CPU Zeit besser aufgeschöpft
- **Highest Response Ratio Next:** Selection function: RR = (w+s) / s → w gesamte Wartezeit bisher, s geschätzte Service Time → Auswahl des Prozesses mit größtem RR → keine starvation, nach gewisser Zeit auch lange Prozesse weit vorne
- **Feeback Scheduling:** Selection Function → je mehr Zeit ein Prozess schon von CPU bekommen hat, desto niedriger Priorität → längere Zeitscheiben
- **First Come First Serve:** FIFO Queue → keine Starvation, dafür möglicherweise langes herauszögern, da vorhergehender Prozess zuerst abgearbeitet warden muss

Round Robin und Virtual Robin brauchen keine Zeitschätzung, wobei der Virtual Round Robin I/O lastige Prozesse nicht benachteiligt
Highest Response Ratio Next und Feeback Scheduling müssen zwar Zeit abschätzen, können so aber einen höheren Durchsatz an kurzen Prozessen erzielen, wobei Highest Response Ratio Next Starvation ganz verhindert

Wir betrachten ein System mit Virtual Memory Management. Diskutieren Sie, welche der folgenden Situationen beim Referenzieren einer virtuellen Adresse auftreten bzw. nicht auftreten kann: (a) TLB Miss ohne Page Fault, (b) TLB Miss mit Page Fault, (c) TLB Hit ohne Page Fault, (d) TLB Hit mit Page Fault. (4)

Page Tables oft sehr groß → verschachtelt → mehrere Hauptspeicherzugriffe notwendig → TLB (Translation Lookaside Buffer) → einige Einträge jedes Page Tables im Hauptspeicher

(a) TLB Miss ohne Page Fault: kann auftreten, da nur bei einem TLB miss im Page Table nachgesehen wird
(b) TLB Miss mit Page Fault: kann auftreten falls nach Zugriff auf Page Table auch dort Frame nicht im Hauptspeicher
(c) TLB Hit ohne Page Fault: Bei einem TLB Hit tritt nie ein Page Fault auf
(d) TLB Hit mit Page Fault: kann nicht auftreten

Mit welcher logischen Struktur des I/O Systems versucht man bei der Realisierung von I/O Funktionen sowohl ein einheitliches Programmierinterface, als auch eine möglichst gerätespezifische Ansteuerung zu erreichen? (4)

Durch ein Schichtmodell, die erste Schicht (Logical I/O) beschäftigt sich mit dem User Input und übergibt diesen an die zweite Schicht (Device I/O), die daraus korrekte I/O Funktionen ableitet und diese dem Scheduler übergibt der dann direkt mit der Hardware kommuniziert. Der erste Abschnitt kann jedoch noch in weitere kleine Abschnitte zerlegt werden, z.B. wenn das Gerät ein File System unterstützt kann ein solche dem User zu Verfügung gestellt werden.

Was versteht man unter Synchronous bzw. Asynchonous I/O? Beschreiben Sie die beiden Arten, I/O-Operationen durchzuführen. (3)

Synchonous I/O: Bei synchronous I/O wir der ausführende Prozess blockiert bis die I/O Operation ausgeführt ist → z.B. Prozess wartet bis Ausgabe am Bildschirm tatsächlich erfolgt ist

Ansynchronous I/O: Hier kann der Prozess falls möglich die Daten an den I/O Buffer übergeben und weiterarbeitn → z.B. Prozess schickt Daten an Bildschirm und arbeitet weiter, auch wenn noch nicht am Bildschirm erschienen

Erklären Sie die Begriffe Access Control List und Capability List. Wozu und wie werden diese verwendet? (4)

Access Control List: Zugriffsrechte bei Objekten gespeichert, Differenzierung nach Benutzergruppen, leichtes Ändern der Zugriffsrechte von Objekten

Capability List: Pro Prozess List mit Objekten und erlaubten Zugriffsoperationen → Tickets regeln Zugriff → Weitergabe und Vererbung von Tickets → Fälschungssicherheit durch Verschlüsselung

Wie berechnet sich die mittlere Zugriffszeit beim Lesen von Daten von einer mechanischen Festplatte? Geben Sie die charakteristischen Zeitparameter einer Festplatte an. Durch welche Strategien kann das Betriebssystem dazu beitragen, die mittleren Zugriffszeiten auf die Festplatte zu reduzieren? (5)

Ta = Ts + Trd + Ttf

Ts = benötigte Zeit um Disk-Arm zur gewünschen Spur zu bewegen (Mittelwert)

Trd: Zeitverzögerung bis Anfang desgesuchten Sektors gefunden (im Mittel: halbe Umdrehungszeit der Disk) → 1/(2*r) (r = Umdrehungsgeschwindigkeit [U/sec]

Ttf: benötigte Zeit zum Übertragen der Daten → b/r*N (b = Anzahl der zu übertragenden Bytes, N = Anzahl der Bytes pro Spur)

Strategien zur Verringerung der mittleren Zugriffszeit:
- Disk Scheduling: FIFO, LIFO, shortest service time first, SCAN (Elevator Alg.), C-Scan, N-step-SCAN und FSCAN
- Disk Caching: Teile der Disk im Hauptspeicher enthalten → ausnutzen von Lokalität

Beschreiben Sie das typische Layout einer Disk bzw. eines Filesystems. Welche Rolle spielen die einzelnen Teile beim Hochfahren des Betriebssystems? (5)

Disk bzw. Filesystem ist in einzelene unabhängige Partitionen unterteilt, Master Boot Record (MBR) in Sketor 0 der Disk → Boot Code, Partition Table (start/end, of partitions, active partition);

Wie ist ein i-node aufgebaut? Welche Informationen enthält er? (4)

Ein i-node pro file, ein file pro i-node → Speichert Flags zum Bestimmen von permissions, Zähler wie viele Einträge im File System auf den inode verweisen, owner id, group id, größe des Files, Speicheradresse, letzter Zugriff, letzter Änderung, letzte Änderung am inode

i-node ist Datenstruktur für jedes File, enthält Fileattribute und Referenzen auf die Blöcke des Files → I-node wird nur im Memory gebraucht, wenn ein File verwendet wird → Nachteil Anzahl der Blockreferenzen pro i-node ist begrenzt → Verwendung indirekter, doppelt und dreifach indiriekter Blöcke

Nennen Sie Design Prinzipien für die Konstruktion von sicheren Systemen. Geben Sie für jede Regel ein Beispiel an. (4)

Open Design: keine Sicherheit durch schwere Verständlichkeit des Codes, Sicherheitssysteme müssen trotz Übersichtlichkeit sicher sein

Default Einstellung: keine Berechtigung → User soll nicht beim einfachen einloggen Admin rechte haben

Least Privilege: so wenige Rechte wie möglich, alle Rechte die nötig sind vergeben

Economy of Mechanisms: Fehlervermeidung durch einfachhalten der Sicherheitsmechanismen, Implementierung möglichst auf niedriger Ebene

Acceptability: System nicht so umständlich, dass Nutzer es nicht nutzen/es umgehen möchte

Überprüfung gegenwärtiger Berechtigungen: nicht einfach Annahme, dass anfängliche Berechtigungen überall gelten

Complete Mediation: Kontrolle aller Zugriffe auf Resourcen → auch Ausnahmesituationen

Wodurch unterscheiden sich die Prozesszustände Ready und Blocked? (2)

Ein Prozess im Zustand blocked wartet auf das eintreteten eines events, z.B. beenden von I/O Operation, ein Prozess im Zustand ready ist prinzipiell lauffähig, hat nur zurzeit keine CPU Zeit zugewiesen bekommen.

Wie unterscheidet sich das Blockierverhalten von Kernel Level Threads und User Level Threads? (2)

ULT: Falls Prozess geblockt wird (z.B. Zeit abgelaufen) geht zwar der Prozess an sich in den Blocked state, die Threads bleiben aber im running state, obwohl ihnen keine Prozessorzeit zugewiesen wird. Nurz wenn innerhalb der Threads eine Aktion folgt die ein blockieren hervorruft werden diese in blocked versetzt. Dadurch kann beim wiedereintritt des Prozesses in den Readyzustand jeder Thread problemlos dort weiterarbeiten wo er war und für den switch zwischen Threads wird kein Mode Switch in kernel mode benötigt → Da jedoch nur auf einem Prozessor laufend kann nur ein Thread auf einmal arbeiten

KLT: Thread Management nicht über Thread library sondern im Kernel. Dieser kann die Threads auch auf verschieden Pozessoren aufteilen und erhält so einen besseren Durchsatz. Nachteil ist jedoch der notwendige Mode Switch bei jedem Thread Wechsel

Für die Lösung des Problems des geregelten Eintritts in einen kritischen Abschnitt werden drei Eigenschaften gefordert. (a) Nennen Sie diese drei Eigenschaften und erklären Sie deren Bedeutung. (b) Wodurch werden die drei Eigenschaften gewährleistet, wenn Semaphore zum Schutz eines kritischen Abschnitts verwendet werden? (5)

- Mutual Exclusion: nur ein Prozess im kritische Abschnitt
- Progress: jeder Prozess muss irgendwann in den kritischen Abschnitt eintreten und darf nicht ewig verzögert werden (keine Starvation)
- Bounded Waiting: nach request für kritischn Abschnitt gibt es nur eine abgeschrankte Anzahl an anderen Prozessen im kritischen Abschnitt

Semaphore sind im endefekt nichts anderes als Integer Variablen, wenn 0 muss der Prozess warten und wird in die Blocked Queue eingereiht, wenn >0 wird nächster Prozess in kritischen Abschnitt geleitet. Dadurch, dass jedes mal nur ein Prozess die Semaphore verringern kann und danach die Überprüfung wieder sattfindet ob >0 Mutual Exclusion, falls kein Deadlock vorliegt (kein Prozess

mehr vorhanden der Semaphore erhöht) kommt durch Queue jeder Prozess irgendwann dran → Progress, Bounded Waiting ergibt sich dadurch, dass meist eine FIFO Queue verwendet wird → maximal so viele Prozesse wie sich davor in die Queue gestellt haben können vor dem Prozess dran kommen

Gegeben sei ein Computersystem, in dem Ihnen zur Synchronisation bzw. Kommunikation von Prozessen nur Nachrichten zur Verfügung stehen (d.h., es gibt keine Semaphore oder andere Synchronisationskonstrukte). Nennen Sie zwei verschiedene Möglichkeiten, wie Sie in diesem Computersystem einen konsistenten Datenaustausch zwischen parallelen Prozessen realisieren können. (4)

Blockierendes und nicht blockierendes Message Passing:

- Blockierend: beim empfangen und senden wird gewartet bis der Kommunikationspartner alles empfangen hat
- Nicht blockierend: Prozess sendet nachricht und arbeitet gleich weiter ohne auf richtiges zustandekommen der kommunikation zu warten

Bei der Realisierung von Dateisystemen gibt es verschiedene Möglichkeiten, um die zu einer Datei gehörenden Datenblöcke zu organisieren bzw. auffindbar zu machen (BlockAllokierung). Nennen Sie vier verschiedene Strategien zur Block-Allokierung von Dateien und beschreiben Sie diese mit ihren Vor- und Nachteilen. (6)

The Pile: Jede kommende Datei wird einfach an den bereits gewählten Platz anghängt → Haufen → Informationen zur Länge und Besonderheiten müssen entweder mitgespeichert werden oder der Deafult-Wert für diese File Art angenommen werden → zum Auffinden eines gewissen Wertes oder Feldes muss gesamter Haufen durchgegangen werden bis gefunden wurde

Sequential File: Jeder Record wird auf Block gleicher Länge gespeichert wobei jedes Feld eine bestimmte Funktion übernimmt → Auffinden einzelner Werte einfacher → besonders gut bei sequentieller Abarbeitung von Dateien, da alles schön angereiht ist → Fixe Größe macht Problem bei sehr großen, sehr kleinen Files und beim Vergrößern bzw. Verkleinern

Indexed Sequential File: Es wird ein Index mit den Key Files und zugehörigem Pointer abgespeichert → Es muss nicht immer gesammter Record geladen werden um auf key zuzugreifen → Overflow File existiert für sehr große Records

Indexed File: Jedes File kann überall gespeichert werden und es werden Indizes angelegt um einerseits auf den Speicherort des gesamten Records zu verweisen oder auf bestimmte Attribute, damit muss nicht immer alles durchgegangen werden falls oft auf einzelne Fleder zugegriffen wird

Direct or Hashed File: Key wird gehasht und freier Zugriff auf alle Speicherort werden ausgenutzt.

Wozu wird die Clock Policy verwendet? Beschreiben Sie deren Funktionsweise. (5)

Die Clock Policy wird verwendet um beim Speichermanagement im Falle eines Page Faults einen Speicherplatz zu finden. Dazu wird immer ein Flag gespeichert und wenn ein Frame verwendet wurde auf 1 gesetzt. Wenn ein Page Fault auftritt wird nacheinander der Zeiger auf den nächsten Frame gesetzt und auf 0 gesetzt bis einer Auftritt der schon auf 0 ist → neuer Speicherort

Was versteht man unter der Working-Set Strategie? Beschreiben Sie deren Funktionsweise und erklären Sie, wie diese Strategie zur Optimierung eines Paging-Systems eingesetzt werden kann. (5)

Es wird ein Tuppel angelegt W(D,t), wobei D die aufgerufenen Frames im Zeitraum t darstellt → wenn Frame nicht mehr im Working Set → kann ersetzt werden → wenn mehr Frames allokiert werden müssten als möglich → probiere später erneut

Beschreiben Sie Aufgabe und Funktion eines Translation Lookaside Buffers? Worauf hat man bei der Betriebssystemimplementierung bei einem Process Switch zu achten, wenn man einen Translation Lookaside Buffer verwendet? (4)

Der TLB wird verwendet um bei Page Tables dafür zu sorgen, dass man bei verschachtelten Page Tables nicht jedesmal alle aus der Sekundär Speicher laden muss → gewisse Page Tables werden im Hauptspeicher gehalten → Falls nicht im TLB (= TLB Miss) → Page Table muss geladen werden → Falls nicht im Page Table erst Page Fault →Bei Process Switch muss auch TLB des zugehörigen Prozesses geladen werden bzw. gespeichert werden

Was versteht man unter Long-Term Scheduling, Mid-Term Scheduling und Short-Term Scheduling? Erklären Sie jeden der Begriffe. (3)

Long-term Scheduling: Die Requests an startenden Prozessen in möglichst effizienter Reihenfolge in die Ready Queue einfügen

Mid-Term Scheduling: Wann möglichst effizient die Prozesse aus der Ready-Suspend Queue wieder in die Ready Queue einlassen

Short-Term Scheduling: Welchen Prozess als nächsten bearbeiten

Erklären Sie die Begriffe Process Switch und Mode Switch, sowie die Beziehung, in der diese beiden Konzepte stehen. Zählen Sie weiters die drei Klassen von Ereignissen auf, die einen Mode Switch nach sich ziehen. (4)

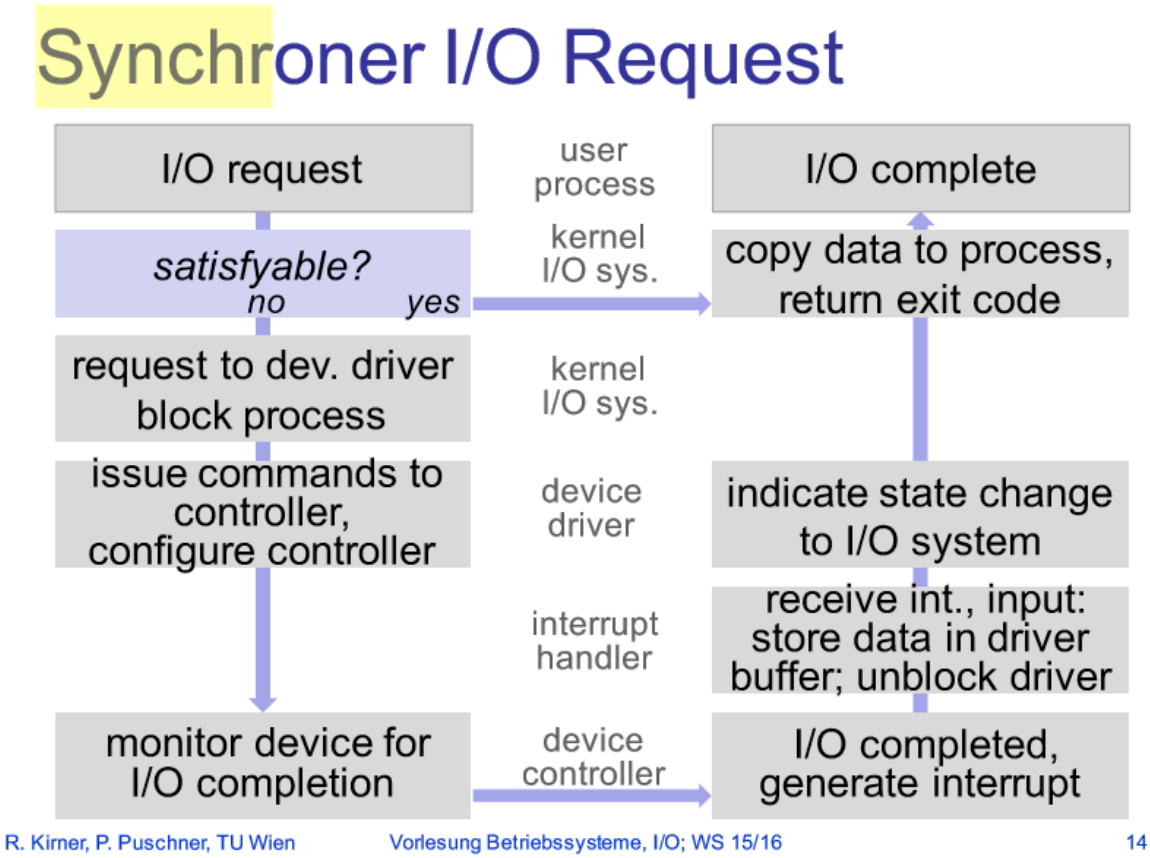
Process Switch: Sichern des Kontextes (Programm-Counter, Prozessor-Register, PSW) des alten Prozesses, ändern seines Zustandes je nach Ursache der Process Switches (in Ready, Blocked, Blocked-Suspend), Einfügen in die jeweilige Queue (Blocked-Queue, Ready-Queue,...). Finden einer Scheduling Entscheidung, d.h. Auswählen eines neuen Prozesses aus der Ready-Queue und seinen Zustand auf Running setzen. Laden des Kontextes des neuen Prozesses in den Prozessor. → Mode Switch auf jeden Fall notwendig

Mode Switch: Wechsel von user mode in kernel mode bzw. umgekehrt (bestimmte Speicherbereiche geschützt vor Zugriff von user mode) → context vom Prozess gespeichert in PCB, nicht notwendigerweise ein process switch notwendig

Was versteht man unter Virtual Memory Management? Welche Vorteile bietet es? (5)

Beim Virtual Memory Management wird, ähnlich wie beim Paging, dynamische Adressübersetzung verwendet → Prozesse werden in Seiten aufgespalten → nicht ganzer Prozess muss im Hauptspeicher sein sondern nur derzeit benötigte Teile → Segmenttabelle für Überprüfung → falls nicht im Hauptspeicher → aus Sekundärspeicher laden → bei hohem Vorkommen → „Thrashing“: Prozessorleistung wird heruntergesetzt durch ständiges laden aus Sekundärspeicher

Skizzieren Sie die Folge der Schritte, die bei der Abarbeitung eines Synchronen I/O Requests ablaufen. (5)



Welche Möglichkeiten kennen Sie, um in einem Paging-System Speicherschutz zu realisieren? (3)

Überprüfung ob Adresse im gültigen Bereich liegt (in der Bound) ???

Was ist Swapping? Wann wird es angewandt? (2)

Das Auslagern eines Prozesses in den Sekundärspeicher → Falls im Hauptspeicher Platz geschaffen werden muss um neue Prozesse einzulagern → z.B. alle Prozesse warten auf I/O Ende, Prozessor unbelastet → Prozesse die warten werden in den Suspend Zustand versetzt und ausgelagert und später wieder zurückgeholt → Zustand wird gespeichert

Nennen Sie die Arten von Optimierungszielen, die ein Scheduler beim Prozess-Scheduling verfolgen kann und geben Sie jeweils Beispiele an. (4)

Durchsatz: desto besseres Scheduling, desto mehr Prozesse können nacheinander beendet werden → Shortest Remaining Time
Prozessorauslastung: möglichst Prozessor immer arbeitend halten und wenig Leerlaufzeiten → Swapping falls möglich → Virtual Round Robin
Fairness: kein Lifelock von Prozessen, alle Prozesse müssen irgendwann beendet werden und sollten nur limitierte Zeit aufgeschoben werden → z.B. Round Robin
Response Time: möglichst schnelle Antwort → z.B. Shortest Process Next
Einahlten von zeitlichen Vorgaben: Alle Prozesse sollten auch spätestens dann beenden wenn gefragt (EDN – Earliest Deadline Next)

Was ist Thrashing, wodurch kommt es dazu? Wie erkennt das Betriebssystem Thrashing? Wie kann dieses Problem beseitigt werden? (4)

Thrashing kommt zustande wenn beim Zerteilen des Prozesses in Seiten ständig nicht im Hauptspeicher vorhanden Seiten nachgeladen werden müssen → ständige Belastung des Prozessors nur um Seiten nachzuladen
Erkennung: mehr Zeit wird zum Laden der Seiten benötigt als zum Ausführen des Prozesses an sich (Resident Set zu klein)
Lösung: Mehr Speicher dem Prozess zuordnen, sonst suspend und zu späterem Zeitpunkt erneut versuchen

Was versteht man unter Blocking bzw. Non-Blocking I/O? Beschreiben Sie die beiden Arten, I/O-Operationen durchzuführen. (3)

Blocking IO: Prozess wird blockiert bis IO Operation abgeschlossen → z.B. Prozess wartet bis Daten auf Bildschirm ausgegeben

Non-Blocking IO: Prozess blockiert wenn möglich nicht sondern übergibt die Daten und arbeitet dann weiter ohne auf Abschluss zu warten → muss nicht in Zustande blocked wechseln

Beschreiben Sie das Prinzip einer Sicherheitsattacke durch Stack/Buffer Overflow. Wodurch kann man sich bei der Implementierung eines Betriebssystems vor einen solchen Angriff schützen? (4)

Durch ausnutzen von nicht Überprüfung von Arraygrenzen wird versucht schadhaften Code in gewisse Speicherbereiche zu kopieren und so anderes Verhalten zu erzwingen bzw. Absturz bestimmter Teile hervorzurufen → Bei jeder Eingabe Grenzen überprüfen (z.B. verwenden von strncpy statt strcpy, da hier nur maximal n Zeichen kopiert werden)

Was versteht man unter einem Process Image? Erklären Sie, aus welchen Teilen ein Process Image besteht? (4)

Besteht aus:

- PCB (Process identification, Processor state information, Process control information)
- User Stack
- Private user address space (programs, data)
- Kernel stack
- optional: Shared address space

Nennen Sie die Bedingungen für das Eintreten eines Deadlocks und erklären Sie diese. (5)

- Mutual exclusion: Nur ein Prozess darf gleichzeitig im kritischen Abschnitt sein
- Hold and wait: Prozesse dürfen Ressourcen halten während sie auf neue warten
- No preemption: Prozesse geben Ressourcen erst wieder frei, wenn sie nicht mehr benötigt werden → können nicht dazu gezwungen werden
- Circular Wait: zyklisches Warten auf Ressourcen, welche anderer Prozess belegt, welcher direkt oder indirekt auf Ressourcen wartet, die der Prozess schon angefordert und wegen no-preemption nicht wieder freigegeben hat

Bei welchen der folgenden Scheduling-Strategien kann es zur Starvation kommen: (a) FCFS, (b) Shortest Job First, (c) Round Robin, (d) Priority Scheduling? Begründen Sie jeweils Ihre Antwort. (4)

- FCFS: Nein, da man maximal warten müssen bis alle zuvor gekommenen abgearbeitet sind → jedoch benachteiligung kurzer Prozesse
- Shortest Job First: Ja, da lange Prozesse möglicherweise nie kürzer als ein anderer sind

- Round Robin: Nein, da jeder Prozess zyklisch eine Zeitscheibe erhält → I/O lastige Prozesse nutzen diese jedoch schlecht aus und verringern damit Effizienz
- Priority Scheduling: ja, da Scheduling außerhalb des Prozessors passiert und falls kurze Jobs öfter mit höherer Priorität versehen werden, kann es zu Starvation bei längeren Prozessen kommen

Was versteht man unter interner Fragmentierung und externer Fragmentierung? Beschreiben Sie die Begriffe und geben Sie je ein Beispiel an. (3)

Interne Fragmentierung: bei z.B. fixer Partitionierung wird oft einem sehr kleinem Speicherbereich eine große Partition zugewiesen → innerhalb der Partition ungenutzter Speicherplatz der nicht vergeben werden kann

Externe Fragmentierung: bei z.B. dynamischer Vergabe des Speicherplatzes kommt es beim Löschen und neu Vergeben einzelner Partitionen dazu, dass zwischen den einzelnen Partitionen kleine Bereich übrig bleiben, die dann nicht wieder gefüllt werden → mehrere einzelne Partitionen verbrauchen Platz und können nicht richtig verwendet werden

Beschreiben Sie den Aufbau und Verwendung einer Multilevel Page Table (eventuell mit Skizze). Warum werden Multilevel Page Tables verwendet? (3)

Multilevel Page Tables falls große Prozesse mit großer Page Table die nicht mehr in Hauptspeicher passt → Unterteilung in weitere Page Tables

Bei der Deadlock-Vermeidung spricht man von einem Safe State bzw. einem Unsafe State. Erklären Sie die Bedeutung dieser Begriffe. (4)

Safe State: es gibt eine Möglichkeit alle Prozesse abzuarbeiten und in den Zustand finished zu bringen → kein Deadlock

Unsafe State: es könnte sein, dass Ressourcenanforderungen der Prozesse erfüllen kann, falls keine freigegeben werden → Deadlock tritt vielleicht auf, jedoch nicht sicher

Wie funktioniert das CPU-Scheduling nach dem Highest Response Ration Next Verfahren? Welche Vorteile bzw. Nachteile hat diese Schedulingstrategie? (4)

Nächster Prozess wird über selection function gewählt → RR = (w + s)/s → s = geschätzte Service Time, w = bereits gewartete Zeit → desto länger ein Prozess wartet, desto höher wird seine Priorität → Vorteile: keine Starvation, kurze Prozesse trotzdem fair behandelt → Nachteile: Schätzung der Service Time notwendig → aufwendig

Erklären Sie die Begriffe Deadlock , Lifelock und Starvation. (4)

Deadlock: durch zyklische Abhängigkeiten beim Zugriff auf Ressourcen, kann kein Prozess die notwendigen Ressourcen anfordern, gibt diese aber wegen no-preemption auch nicht wieder ab, sodass ein anderer Prozess darauf zugreifen kann

Lifelock: Prozess wird Eintritt in kritischen Abschnitt nie gewährt → kann so keinen Fortschritt machen

Starvation: Ein Prozess kann auf bestimmte Ressourcen nicht zugreifen, weil immer zuerst andere Anfragen abgearbeitet werden → kann auch nicht weiter im Ablauf da die Ressourcen fehlen

Jeder Prozess benötig für seine Abarbeitung CPU und Arbeitsspeicher. Welche Strategien werden angewandt, um das Auftreten eines Deadlocks durch den Wettbewerb um diese Ressourcen auszuschließen? Erklären Sie, wie diese Verfahren auf einem Computersystem konkret realisiert werden.

Deadlock prevention: Deadlock Bedingungen werden versucht aufzuheben → Hold and Wait könnte aufgehoben werden, wenn jeder Prozess alle seine Ressourcen schon zu Beginn anfordern müsste und erst dann anfangen dürfte zu arbeiten, no preemption könnte durch die Erlaubnis, dass Ressourcen während des Prozesses auch wieder frei gegeben werden müssen aufgelöst werden und Prozess zurückgesetzt wird → kompliziert zu realisieren
Circular wait kann dadurch verhindert werden, dass man jeder Art der Anforderung einen Wert zuweist, mit laufender Zeit nur Ressourcen mit immer höherer Ordnung anforderbar → da immer größer sein muss kann nur eindeutige Ordnung passieren

Deadlock avoidance: Ein Prozess wird entweder gar nicht erst gestartet, wenn bei Ressourcenanforderungen ein Deadlock auftreten könnte →oder Ressourcenanforderung wird nicht gewährleistet wenn Deadlock auftreten könnte → Banker's Algorithm

Deadlock detection: Nach jeder Ressourcenanforderung wird geschaut ob ein Deadlock aufgetreten ist → entweder alle Prozesse abbrechen oder auf Deadlock freien Zustand zurücksetzen oder nur nach der Reihe Prozesse abbrechen bis Deadlock aufgehoben

Eine Festplatte hat 5000 Zylinder, die von 0 bis 4999 nummeriert sind. Es wird gerade auf Zylinder 195 zugegriffen, der vorhergehende Zugriff erfolgte auf Zylinder 164. Requests auf folgende Zylinder stehen zur Behandlung an (in FIFO Reihenfolge angegeben): 92, 1470, 917, 1841, 967, 1518, 1154, 1620, 173. Über wieviele Zylinder muss sich der Lese/Schreibkopf von der aktuellen Position in Summe bewegen, um die Requests nach folgenden Strategien abzuarbeiten: (a) FCFS, (b) Elevator Algorithm (SCAN) und (c) C-SCAN? (6)

FCFS: 103+1378+553+924+874+551+364+466+1447=6660

SCAN: 722+50+237+316+48+102+221+1668+81=3445

C-SCAN: 722+50+237+316+48+102+221+1841+92+81=3710

Wie funktioniert Round Robin Scheduling? Welchen wichtigen Parameter gibt es bei diesem Verfahren? Wie wird man diesen Parameter günstiger Weise wählen? (4)

Beim Round Robin Scheduling wird nacheinander an jeden Prozess eine Zeitscheibe verteilt → alle gleichberechtigt → Parameter der Länge der Zeitscheibe wählbar → Wahl so, dass Process Switches nicht zu häufig auftreten, aber auch nicht so lange, dass I/O lastige Operationen den Prozessor zu wenig auslasten → gute Strategie Scheibe so zu wählen, dass sie die Länge der Ausführung einer typischen Prozessinteraktion etwas überschreitet