(Grundlage Zusammenfassung „TU_Wien-Betriebssysteme_VO_(Puschner)_-_Zusammenfassung_2020.pdf“)

1. Process Control Block

Charakterisiert Prozesse eindeutig, besteht aus:
Process Identification:
- Prozessnummer
- Benutzerkennung
- parent process identifier

Process State Information:
- Registerinhalte (Kontroll- , Statusregister)
- Stackpointer

Process Control Information:
- Querverweise auf andere Prozesse
- Scheduling
- Zusatzinformationen zu dem Prozess

2. Translation Lookaside Buffer

Wird im Kontext des Paging als high-speed-buffer für Page-Table-Einträge verwendet. Dabei ist das Hauptziel die doppelte Speicherzugriffszeit zu reduzieren (Zugriff auf Page-Table-Eintrag erfordert zusätzlich laden der Daten aus dem Hauptspeicher oder Virtual Memory). Kürzlich aufgerufene Page-Table-Einträge werden dementsprechend im TLB gespeichert (funktioniert analog zum Memory Cache).

3. Memory Management bei Prozessen

Hauptziele: Partitioning, Relocation, Protection, Sharing, Performance

Dynamic Partitioning: Speicherblöcke für Prozesse werden dynamisch anhand der Prozessgröße erzeugt (first-fit, best-fit, next-fit), externe Fragmentierung kann auftreten (bei Prozessbeendigung enstehen Lücken zwischen den Speicherblöcken, welche nicht effizient nutzbar sind → Lösungsstrategie Compaction; Zusammenschieben der Lücken)

Fixed Partitioning: Speicherblöcke haben fixe Größen, zu große Prozesse müssen auf mehrere Blöcke aufgeteilt werden, interne Fragmentierung kann auftreten (Partition wird ineffizient genutzt, da ein Prozess mit wenig Ressourcen den ganzen Block blockiert), Bei einer Queue pro Partition kann es zu ineffizienter Nutzung kommen

bei Prozessen derselben Größe → eine Queue für alle Partitionen verwendet und optimale Ausnutzung anstreben

4. Erklären, was das Roundrobin-Scheduling Konzept ist

Verfahren zur Auswahl des nächsten Prozesses. Dabei werden Prozessen nacheinander Zeitscheiben zugewiesen und diese werden schrittweise abgearbeitet (arbeiten nach FCFS, preemptive)→ führt zur Benachteiligung von I/O-Prozessen (kann Ressourcen nicht voll ausschöpfen). Lösungsstrategie → Virtual Round Robin (VRR): Es existiert zusätzlich eine „auxiliary queue“, in welcher priorisierte I/O-Prozesse stehen.

5. Was sind Deadlocks?

Zustand in dem mindestens 2 Prozesse auf die gegenseitige Freigabe von Ressourcen warten. In diesem Fall müssen 4 Bedingungen gelten: Mutual Exclusion, Hold & Wait, Circular Wait, No-Preemption

6. Threads

- Threads sind Teile von Prozessen, welche unabhängig ausführbar sind
- Haben eigenen User- , Kernelstack
- Greifen gemeinsam auf Informationen von dem zugehörigen Prozess zu
- können in 2 Modi ausgeführt werden: User-Level-Thread, Kernel-Level-Thread

7. ULT – KLT

ULT:

- Synchronisation zwischen Threads muss im ULT muss vom Programmierer gewährleistet sein (mittels Thread-Libraries)
- Threads nicht über mehrere Prozesse synchronisierbar
- Kernel muss nicht eingeschaltet werden

KLT:

- Process Switch über Prozesse hat 2 Mode-Switches
- Kommunikation läuft über Kernel
- Vorteil

8. Microkernel

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 dem 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

9. Virtual Memory
- Wird verwendet um den Hauptspeicher zu erweitern
- Beim Swapping (ist I/O-Prozess) werden Daten hier ausgelagert
- Wird beim Paging (Page Faults) verwendet

10. logische auf physikalische Addresse umwandeln (Page Table)
Mittels der Page-Nummer, Offset und dem Page-Table-Pointer wird in der Page-Table nach der logischen Adresse geschaut. Bei dem passenden Eintrag existiert eine dazugehörige Frame-Number. Diese wird dann anschließend mit dem Offset zusammengefügt. Dadurch erhält man die physikalische Adresse.

11. Thrashing
Verbringt der Prozessor zu viel Zeit damit Pages von der Virtual Memory zu laden kann es zum Thrashing kommen, bei dem ein drastischer Einbruch der Effektivität und eine starke Verlangsamung der Abarbeitungsgeschwindigkeit eintritt.

12. Working Set Strategie
Working-Set sind alle aktiven Pages eines Prozesses. Anhand einer Funktion W(D,t) wird schrittweise immer wieder die Aktivität von Frames überprüft und inaktive Pages werden dementsprechend aus dem Working-Set gelöscht.

13. Deadlock Bedingungen
Mutual Exclusion:
Ein Prozess hat exklusiven Zugriff auf Ressourcen.

Hold & Wait:
Prozess darf Ressourcen trotz Wait behalten.

No-Preemtpion:
Die einem Prozess zugewiesenen Ressourcen dürfen nicht weggenommen werden.

Circular wait:
Zyklische Abhängigkeiten von Prozessen bezogen auf Ressourcenfreigabe.

14. Deadlock Prevention genauer

Systemdesign welches versucht, die Entstehung von Deadlocks zu umgehen.
- Mutual Exclusion kann nicht vermieden werden
- Hold & Wait: Prozess blockiert so lange, bis alle Ressourcen angefordert werden können
- No Preemption: Prozess wird zur Freigabe von Ressourcen gezwungen
- Circular Wait: strikte lineare Ordnung für Ressourcenklassen

15. Deadlock Avoidance genauer

Process Initiation Denial:
- Prozess wird erst gestartet, wenn alle Ressourcen angefordert werden können
- Neuer Prozess wird nur gestartet, wenn die Summe aller bisherigen maximalen Anforderungen und die Anforderung des neuen Prozesses kleiner ist, als die gesamten verfügbaren Ressourcen. (Claim-Matrix → Anforderungsmatrix)

Ressource Allocation Denial (Banker's Algorithm):
- Ressourcenanforderung wird nur gewährt, wenn kein Deadlock entstehen kann
- Need-Matrix = Claim-Matrix – Allocation-Matrix
- (un)safe state: Es existiert mindestens eine Anforderungsreihenfolge, die zu (k)einem Deadlock führt
- Es wird die Need-Matrix mit dem V-Vektor (alle noch freien Ressourcen) verglichen

16. Datenstrukturen Prozesse (Process Image, PCB, …)

Ein Prozess ist ein Programm während der Laufzeit (dynamisch).
Process Image (read-only) besteht aus:
- User Data
- User Program
- Systemstack
- PCB

17. Scheduling

Scheduling beschreibt Auswahlverfahren nach denen Prozesse ausgewählt werden, unter folgenden (Optimierungs-)Zielen:
- Durchsatz
- Prozessorauslastung
- Response Time
- Einhaltung zeitlicher Vorgaben

3 Arten von Scheduling:
- long-term-scheduling: LTS-Scheduler selektiert Programmanfragen
- mid-term-scheduling: MTS-Scheduler ist Teil der swapping-Funktionalität
- short-term-scheduling: STS (Dispatcher) wählt nächsten Prozess aus

18. Round Robin vs FCFS

- FCFS ist preemptive, Round Robin non-preemptive
- lange Prozesse in FCFS blocken die restlichen Prozesse
- Round Robin basiert auf FCFS
- Round Robin grundsätzlich besser geeignet

19. Highest Response Ratio Next (Was wird dadurch erreicht?)

Der nächste Prozess wird anhand einer Formel bestimmt: RR = (w+s) / s
Dieser Scheduling-Algorithmus verhindert Starvation und bevorzugt kurze (und lang wartende) Prozesse. Non-Preemptive

20. Paging-Strategien (LRU vs. FIFO)

LRU (Least Recently Used):
Die am wenigsten verwendeten Pages werden ersetzt, an sich gute Strategie verursacht aber viel Overhead (da Zeitstempel gespeichert werden müssen)

FIFO (First in first out):
Oft verwendete Pages müssen auch dementsprechend oft erneut geladen werden.

21. Clock-Policy

Den Page-Frames wird ein zusätzliches Bit (Use-Bit) hinzugefügt, welches im Kreis (Frame-Pointer) abgefragt wird. Ist das Use-Bit nicht gesetzt (d.h.0) wird diese Page ersetzt und bei allen bisherigen durchwanderten Page-Frames wird das Use-Bit auf 0 gesetzt. Bei dem neuen Page-Frame hingegen auf 1.

22. Bedingungen kritischer Abschnitt
- wechselseitiger Ausschluss (Mutual Exclusion)
- Atomizität
- Bedingungssynchronisation

23. Definition von Init/P/V
I….Initialize (>=0)
P….passeren (Sem Wait; vermindert counter um 1; blockiert <=0)
V….verhogen (Sem Post; erhöht counter um 1)

24. Replacement-Strategien
OPT-Policy:
- Page wird ersetzt, dessen Referenz am weitesten in der Zukunft liegt (theoretisches Konzept, nicht implmenetierbar)
- Wird zur Bewertung anderer Strategien verwendet

LRU, FIFO (Siehe Frage 20)
Clock-Policy (Siehe Frage 21)

25. Buffering
- Buffer ist Zwischenspeicher für I/O-Operationen
- Um Effizienz zu erhöhen, wird zuerst in den Buffer geschrieben und erst nach einer Weile wird der Buffer ausgeschrieben, um weniger I/O-Zugriffe zu haben.
- Single Buffering: Nur 1 Buffer wird ausgewertet
- Double Buffering: Ein Buffer wird ausgewertet, der andere kann währenddessen beschrieben werden
- Circular Buffering: Wie double-buffering nur mit mehr Buffern

26. Chained/Continuous/Index Allocation und I-Nodes
Chained Allocation:
Blöcke sind durch Pointer verbunden und dadurch können Daten auf mehreren Blöcken verteilt sein. Das Problem hierbei ist, dass Pointer auch Speicher mitbelegen.

Continuous Allocation:
Blöcke sind miteinander verbunden. Dies führt jedoch zu externer Fragmentierung. Verwendung bei CD/DVD-Roms

Indexed Allocation:

Pointer werden extra im FAT (File Allocation Table) abgespeichert, jedoch verbraucht dieser viel Arbeitsspeicher.

I-Nodes:
Eigene Datenstruktur, welche File Attribute und Referenzen auf die Blöcke des Files enthält. Kann auch indirekte Verweise haben (Filetabellen).

27. Security Objectives

Confidentiality:
Vertraulichkeit, dabei ist wichtig, dass nur autorisierte Personen Read-Zugang haben.

Integrity:
Daten dürfen nicht unerlaubt manipuliert werden.

Availability:
Daten nur dort verfügbar wo benötigt.

28. Capability Lists/ACL

Access Control List (ACL):
Zugriffsrechte werden bei den Objekten gespeichert → leichte Änderung möglich

Capability Lists:
Pro Prozess gibt es eine Liste an Objekten und Zugriffsrechte sind in Form von Tickets vorhanden → Verebung der Zugriffsrechte leicht möglich

29. Wie funktioniert das File System in UNIX (i-Nodes)?

In Linux Systemen enthält jede File eine I-Node in der zum einen Attribute und Informationen gespeichert werden die Auskunft über Permissions, Owner, Größe, etc. geben sowie Referenzen die auf die Daten der Datei im Speicher zeigen. In UNIX-Systeme sind I-Nodes im Superblock → erste I-Node verweist auf Root-Directory.

30. In welchen Schichten ist I/O organisiert?

- Logical I/O (stellt das Interface mit den Funktionen dar)
- Device I/O (Operationen werden in Gerätekommandos übersetzt)
- Scheduling & Control (Verarbeitung/Verwaltung von Gerätekommandos, Interaktion mit Hardware)

31. Deadlock Detection

- Deadlock wird in periodischen Abständen geprüft
- Falls ein Deadlock vorherrscht, wird eine Recovery-Strategie benötigt (zB. Rollback, schrittweiser Ressourcenentzug, Abbruch)

32. Inverted Page Table

Bei dieser Art der Organisation von Page-Tables existiert eine Page-Table für den gesamten Arbeitsspeicher. Zusätzlich gibt es einen Eintrag pro physikalischem Frame.

33. Multi-Level Page Table

Innerhalb einer Page-Table gibt es eine Root-Page-Table, welche als Verzeichnis für weitere Page-Tables dient. Dieser Fall tritt ein, wenn die Page-Table im Laufe der Zeit immer größer wird.

34. Bell and LaPaluda

- Modell für den Informationsfluss.
- Bestimmt Hierarchie von Security-Klassifikations
- Security Axioms

read: SC(S) >= SC(O) („no read up“)
append: SC(S) <= SC(O) („no write down“)
read-write: SC(S) = SC(O)

35. Semaphoren

Sind ein Synchronisationsmechanismus zur Regulierung des Zugriffs auf gemeinsame Ressourcen (verzicht auf busy waiting). Dabei wird eine counter-Variable entweder um 1 erhöht (semSignal) oder um 1 vermindert (semWait). Ist der Wert der Variable <=0 blockiert semWait die weitere Prozessausführung.

36. Relocation

Wird zur dynamischen Positionierung von Prozessen verwendet. In diesem Zusammenhang existieren 3 unterschiedliche Adressarten:
- logische Adresse (unabhängige Referenz auf eine Position im Speicher)
- physikalische Adresse (tatsächliche Position im Hauptspeicher)
- relative Adresse (Adresse in Bezug auf einen bestimmten Punkt)

37. Buddy System

Der Speicherplatz wird in 2er-Potenzen aufgeteilt bis der kleinstmögliche Bereich für den Prozess erreicht wurde. Anschließend kommen weitere Prozesse in den passenden des Speichers, wobei alle Bereiche aufgrund der Zerteilung die Größe einer 2er Potenz haben.

38. Process Switch

Ablauf eines Prozesswechsels:
- Umschalten des aktiven Prozesses (Speicherung des PCB, Register, Stack um Fortsetzen des Prozesses zu ermöglichen)
- Alten Prozess entweder auf Blocked-Queue setzen oder terminieren
- Neuen Prozess aus Ready-Queue holen
- Neuen Prozess der CPU zuteilen

Ereignisse die zu einem Process Switch führen:
- Supervisor Call (Aktivierung von OS)
- Trap (Fehler, Exception)
- Interrupt (Fehler außerhalb des Prozesses, Kontroller an Interrupt Handler)

39. Producer-Consumer

Datenaustausch zwischen Producer (generiert Daten) und Consumer (möchte Daten lesen). Das Problem hierbei besteht darin, dass garantiert werden muss, dass der Consumer nur konsumiert, wenn der Producer tatsächlich schon Daten generiert hat. Um dies sicherzustellen wird ein zusätzlicher Synchronisationsmechanismus (neben Mutex) benötigt, welcher gewährleistet, dass dieser Umstand tatsächlich eintrifft (zB. Semaphoren, Monitor).

40. Disk Scheduling (wozu, Verfahren erklären)

Wird verwendet, um Effizienz und Flexibilität in einem OS bei I/O-Operationen zu gewährleisten. Es gibt 4 mögliche Disk-Scheduling Strategien:
- SSTF (Operation mit geringster Bewegung des Lesearms wird gewählt)
- SCAN (Lesearm wird in beide Richtungen bewegt)
- C-SCAN (Lesearm nur in eine Richtung, zurück „Leerfahrt“)
- N-step-SCAN (Request-Queue Zerteilung auf Subqueues der Länge N)
- FSCAN (2 Subqueues, einer bearbeitet Requests, anderer wird abgearbeitet)