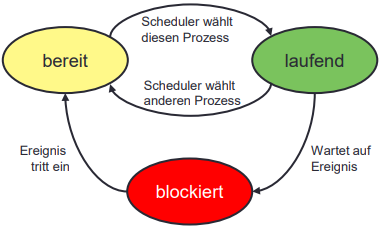
Übungsaufgaben zur Vertiefung der Aufgaben von Betriebssystemen und zur Klausurvorbereitung im Kurs 5CS-OPSY-30

Aufgaben:

1. Betriebssysteme arbeiten zur Realisierung ihrer Aufgaben mit dem Prozessmodell. Was ist das wesentliche Ziel des Prozessmodells? Wie erreicht das Betriebssystem dieses Ziel? Was sind die wesentlichen Bestandteile eines Prozesses?

Skript Betriebssysteme 2, Folien 2-9

* Abstraktion zentraler PC-Ressourcen
* gefühlt gleichzeitige Ausführung von Programmen in einer einzelnen CPU
* jegliche ausführbare Software wird als Menge sequenzieller Prozesse verstanden
* Scheduler steuert und überwacht CPU-Zeit und verteilt diese gerecht und effizient (m.H. einer Prozesstabelle)
* Bestandteile: privaten Adressraum, Befehls- & Programmzähler, Befehlsregister, Prozesstabelle, evtl. Kindsprozesse, Threads



1. Nennen Sie die drei Zustände, die zugelassene und nicht beendete Prozesse in einem Betriebssystem einnehmen können. Skizzieren Sie die Übergänge, die zwischen diesen Zuständen auftreten können

Folien 14-15

1. Können Prozesse logisch noch weiter unterteilt werden, und wenn ja wie?

Folien 26-28

* Threads sind den Prozessen untergeordnet und sind als parallele Verarbeitungsstränge zu verstehen
* Threads eines Prozesses können auf einen gemeinsamen Adressraum zugreifen
* besitzen eigenen Stack, Befehlszähler, Registerwerte und Zustände

1. Was sind Race Conditions und welche Bedingungen führen dazu?

Folien 39-42

* nebenläufige Prozesse oder Threads greifen gleichzeitig auf gemeinsamen Speicher zu
* Endergebnis kann dadurch ein falsches sein, abhängig von der Reihenfolge der Abarbeitung
* Bedingungen:
  + Nebenläufige Prozesse oder Threads werden ausgeführt
  + nutzen gemeinsame Daten

1. Was ist eine kritische Region und mit welchem grundsätzlichen Konzept lassen sich mehrere kritische Regionen gleichzeitig vermeiden?

Folien 42-43

* Zeit während eines Zugriffs auf gemeinsame Daten heißt kritische Region
* Prozess oder Thread muss, während ein anderer Prozess oder Thread in einer kritischen Region arbeitet, daran gehindert werden, in diese einzutreten

1. Erläutern und skizzieren Sie die Funktionsweise einer Sperrvariablen und wie Prozesse mit ihr umgehen, um Race Conditions zu vermeiden. Welches Problem mit Sperrvariablen löst der TSL-Maschinenbefehl, welches bleibt bestehen?

Folien 47-49

* beim Eintritt in kritische Region wird Sperrvariable gesetzt, wodurch anderen P/T signalisiert wird, dass sie nicht ebenfalls eintreten können, diese müssen aktiv warten
* die Test, Set and Lock (TSL) Operation ist nicht atomar, wodurch der Prozess vom Scheduler währenddessen abgeschaltet werden kann und die Werte der variablen überschrieben werden können
* durch den TSL-Maschinenbefehl wird dieses Problem gelöst, das aktive Warten bleibt jedoch nötig

1. Welchen Vorteil besitzen Semaphoren gegenüber Sperrvariablen?

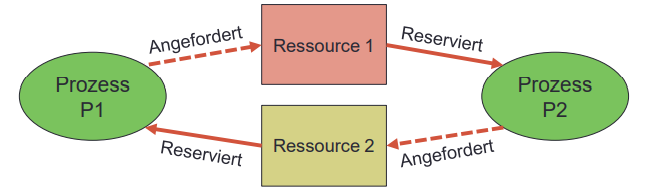
Folien 50-54

* binäre Semaphoren funktionieren ähnlich wie Sperrvariablen, werden aber nicht vom Prozess selbst gesteuert, sondern mit Hilfe von Systemaufrufen
* diese sind atomar und Prozesse, die auf die kritische Region warten, werden blockiert und müssen nicht aktiv warten

1. Stellen Sie eine einfache Deadlock-Situation grafisch dar und erläutern Sie anhand der Darstellung, was ein Deadlock ist

Folien 59

* eine Menge von Prozessen befindet sich im Deadlock, wenn jeder Prozess auf eine Ressource wartet, die nur von einem Prozess derselben Menge freigegeben werden kann



1. Mit welchem Algorithmus lassen sich Deadlocks sicher vermeiden (Deadlock Avoidance)? Erläutern Sie dessen Funktionsweise stichpunktartig. Warum wird dieser Algorithmus nicht praktisch eingesetzt

Folien 76-78

* Bankier-Algorithmus
* benötigt Informationen:
  + R Vektor der insgesamt vorhandene Ressourcen jedes Typs
  + Z Vektor der aktuell zugewiesenen Ressourcen jedes Typs
  + V Vektor der verfügbaren Ressourcen jedes Typs
  + Matrix aller Prozesse
  + Matrix aktuell zugewiesener Ressourcen
  + Matrix der noch maximal benötigten Ressourcen jedes Typs
* 1. Suche Prozess, dessen maximaler Ressourcenbedarf <= V ist
* 2. weise diesem alle noch benötigten Ressourcen zu und lasse ihn Arbeit abschließen. Als beendet markieren, freien Ressourcen wieder zu V hinzufügen
* 1. & 2. wiederholen für alle noch verbleibenden Prozesse als beendet markiert sind, passiert dies ist das System safe, andernfalls unsafe
* praktisch nicht in Benutzung, weil
  + kaum ein Prozess vorher weiß, wie viele Ressourcen er benötigt
  + Anzahl der Prozesse sich im laufenden Betrieb ständig ändert
  + Ressourcen unbemerkt ausfallen können

1. Erläutern Sie stichpunktartig die Funktionsweise des Scheduling-Algorithmus Shortest Job First. Welchen Unterschied gibt es zu Shortest remaining Time Next?

Folien 98-99

* ist nicht unterbrechend
* sortiert die Warteschlange nach der angenommenen Bearbeitungszeit eines Prozesses
* kurze Prozesse kommen also zuerst dran
* Voraussetzung: alle Prozesse gleichzeitig verfügbar, Durchlaufzeit vorher bekannt
* Shortest Remaining Time Next ist unterbrechend
* jeder neue Prozess wird mit dem aktuellen verglichen
* ist der neue kürzer, wird dieser vorgezogen
* bevorzugt neue, kurze Prozesse

Was unternimmt das Betriebssystem beim Swapping? Welches sind die wesentlichen Nachteile des Swappings und was kann dagegen unternommen werden?

Folien 126-129

* Prozesse, die längere Zeit nicht benötigt werden, werden zeitweise aus dem RAM komplett auf die Festplatte geschrieben
* freiwerdender RAM kann dann neuen Prozessen zugeteilt werden
* wird der Prozess wieder benötigt, wird er von der Festplatte erneut in den RAM eingelagert
* Nachteile:
  + Fragmentierung des RAMs
  + Ineffizienz durch hohe Datenmengen und lange Wartezeiten bei Aus- und Einlagerung
* Lösung:
  + vor der Einlagerung von Prozessen sucht das BS eine passende Lücke mit Algorithmus
  + First Fit: Sucht die erste ausreichend große Lücke
  + Best Fit: Sucht die am besten passende Lücke
  + Worst Fit: Sucht die größte Lücke und lässt den größten Restbereich
  + Quick Fit: Findet häufige Größen (4K, 8K, 16K, 20K,…) sehr schnell
  + 🡪 Fragmentierung kann nicht völlig vermieden werden
* nur durch echte Virtualisierung des Speichers möglich
* 🡪 Paging

1. Was ist Paging in der Speicherverwaltung und welche Hardware übernimmt diese Funktion?

Folien 132-141,

* Memory Management Unit (MMU)
* in CPU integriert
* logischer Adressraum der Prozesse wird vom physikalischen Speicher getrennt
* es gibt nur noch virtuelle Adressen, die von der MMU übersetzt werden in die physikalischen Adressen
* virtueller Adressraum wird in Pages unterteilt
* Pages sind zusammenhängende Adressblöcke von fester Größe
* Pages werden auf Page Frames im physikalischen Speicher abgebildet (gleich groß)
* 🡪 Prozesse müssen nicht komplett ausgelagert werden, sondern nur nicht benötigte Prozessteile

1. Vergleichen Sie die Seitenersetzungsstrategien FIFO und Clock-Algorithmus. Welche Vorteile bietet der Clock-Algorithmus?

Folien 157-159

* FIFO ersetzt die älteste Seite
* Alter sagt jedoch nichts darüber aus, ob sie nicht sofort wieder benötigt wird
* einfacher Algorithmus, der aber auch wichtige Seiten entfernen kann
* Clock-Algorithmus (effiziente Variante des Second-Chance-Algorithmus) geht eine ringförmige List nacheinander durch, es wird die älteste Seite gesucht und überprüft, ob das R-Bit (Kennzeichnet Seitenzugriff) 0 ist
* wenn ja kann Seite ersetzt werden
* wenn R-Bit = 1, dann wird es auf 0 gesetzt und beim nächsten Durchlauf nochmal geprüft
* durch ringförmige Liste muss die Reihenfolge der Einträge nicht geändert werden

1. An welcher Stelle einer Partition findet ein Rechner den Programmcode zum Laden eines Betriebssystems? Wo befindet sich im Dateisystem NTFS das „Inhaltsverzeichnis“ mit den Dateiattributen?

Folien 195-196

* in einer als aktiv markierten Partition
* diese enthält ausführbaren Code im Boot-Block zum Laden eines BS
* Inhaltsverzeichnis mit den Dateiattributen in der Master File Table (MFT)

1. Die Blockgröße in Dateisystemen hat Einfluss auf die Effizienz der Speichernutzung auf dem physischen Datenträger. Für welche Dateigrößen sind eher kleine Blöcke, für welche eher größere Blöcke wirtschaftlicher?

Folie 200, tranfer

* bei vielen kleinen Dateien sind auch kleine Blöcke ratsam, um die Blöcke weitestgehend vollständig zu besetzen
* bei vielen großen Dateien (z.B. Videos) sind größere Blöcke ratsam, um die Geschwindigkeit zu erhöhen

1. Erläutern Sie die Funktionsweise eines Image Backups. Welches ist die kleinste physische Datenträgereinheit, die mit einem Image Backup gesichert werden kann?

Folie 219, transfer

* sichert jeden Sektor eines Datenträgers und schreibt diese in eine Image-Datei
* die Image-Datei kann meist gemountet und wie ein normaler Datenträger nach Dateien durchsucht werden
* Datenträger können damit alternativ auch komplett wiederhergestellt werden
* kleinste Datenträgereinheit: Partition

1. Was unternimmt eine CPU, wenn ein hoch priorisierter Hardware-Interrupt auftritt? Warum ist dies in Multiprocessing-Systemen wesentlich effizienter als Polling bzw. programmierte Ein- und Ausgabe?

Folien 229-230

* CPU beendet die Arbeit am aktuellen Prozess und startet den Ein-/Ausgabeprozess für das Gerät
* der neue Prozess und die Geräteanforderungen werden abgearbeitet
* danach wird zurück zum Ursprünglichen Prozess gewechselt
* beim Polling bzw. der programmierten Ein- und Ausgabe fragt die CPU das Gerät zyklisch ab, wodurch viel CPU-Zeit in der Warteschleife verbraucht wird

1. Welche logischen Funktionen übernimmt ein Gerätetreiber und an welcher Position im Schichtenmodell ist er angesiedelt?

Folien 236-237, Betriebssysteme 1 Folie 147

* Treiber sind Software, die gerätespezifische Informationen und Funktionen für das BS zur Verfügung stellen
* sie ermöglichen somit die Kommunikation zwischen BS und Hardware
* Treiber verarbeiten die Interrupt-Anforderungen der Hardware
* die Treiber liegen in den Schichten zwischen der Hardware und den Input/Output Komponenten des BS

1. Warum ist eine grafische Benutzeroberfläche mit sehr viel mehr Aufwand und Rechenleistung verbunden als die zeichenorientierte Ausgabe einer Shell?

Folie 252

* Terminalbildschirme müssen nur eine Matrix aus rund 1000 Zeichen darstellen
* um den Bildschirminhalt einer grafischen Oberfläche darstellen zu können, ist der Bildschirminhalt ein Bitmap-Bild
* um ein Bild in der kleinsten IBM-PC Auflösung (640x360) darzustellen, werden für eine monochrome Darstellung 307200 Bit benötigt

1. Welche zwei Eigenschaften zeichnen Multiprozessorsysteme aus und was unterscheidet symmetrisches von asymmetrischem Multiprocessing?

Folie 264, 266-267

* Multiprozessorsysteme sind Rechner mit mehr als einer CPU, die sich einen gemeinsamen Speicher teilen
* bei symmetrischen Multiprozessorsystemen sind die Prozessoren gleichrangig
* bei asymmetrischen hingegen werden die CPUs ungleichrangig behandelt

1. Beim symmetrischen Multiprocessing ergibt sich ein Problem bei der Behandlung von Interrupts. Worin besteht das Problem und mit welchen Bausteinen und Konzepten wird es behoben?

Folien 278-279

* Maschinenbefehle zum Sperren und Erlauben wirken je Prozessor
* dies muss bei der Synchronisation berücksichtigt werden

1. Mit welchen Konzepten kann auf symmetrischen Multiprozessormaschinen das Problem umgangen werden, dass bei jedem Kontext-Wechsel zwischen Prozessoren deren lokale Cache-Inhalte obsolet werden und neu eingelesen werden müssen?

Folien 289-291

* Space Sharing
* zusammengehörige Threads werden gruppiert und Prozessorgruppen zugeordnet
* darin laufen sie parallel ab
* die Prozessorgruppen befinden sich in einer CPU-Partition

1. Welche Vorteile bietet Hardware-Virtualisierung, welche Nachteile sind mit ihr verbunden?

Folien 299-301

* Vorteile:
  + ein physischer Rechner kann mehrere virtuelle Maschinen unterstützen und so die vorhandene Hardware besser ausnutzen
  + Auslastungsschwankungen können besser ausgeglichen werden
  + Hochverfügbarkeit durch Online-Migration der VMs zwischen physischen Servern
  + historische Applikationen und BS können auf einer VM installiert werden, obwohl sie von der eigentlichen Hardware nicht mehr unterstützt werden
* Nachteile
  + Overhead durch Wirtssystem und die Notwendige Koordination der Gastsysteme
  + Konkurrenz der Gastsysteme um die verfügbare Hardware
  + umfangreichere Auswirkung bei Ausfall eines physischen Servers 🡪 komplexes Management der VMs und zusätzliche Redundanzen erforderlich
  + hohe Komplexität der Datensicherung

1. Worin unterscheiden sich Paravirtualisierung von Vollvirtualisierung und welche Vorteile bietet Paravirtualisierung? Warum wird dann nicht nur Paravirtualisierung eingesetzt?

Folien 314-315, weil bei vielen OS der Quellcode nicht offen ist und daher nicht angepasst werden kann

* bei Vollvirtualisierung ist gesamte Hardware virtualisiert und für das Gastsystem transparent
* bei der Paravirtualisierung werden die Gastsysteme portiert, wodurch deren Kernel an das Laufen auf einem Hypervisor angepasst wird
* das Gastsystem kommuniziert mit dem Hypervisor nicht über Hardwareaufrufen, sondern über spezielle APIs
* somit wird der Hypervisor zum Mikrokernel
* Paravirtualisierung wird nicht immer eingesetzt, weil bei vielen BS der Quellcode nicht offen ist und deshalb nicht angepasst werden kann

1. Durch welche konzeptionelle Änderung bietet die Hardwareunterstützte Vollvirtualisierung Vorteile gegenüber der herkömmlichen Vollvirtualisierung?

Folie 310-311, CPU erlaubt dem Hypervisor Kernel-Modus-Befehle

* privilegierter Befehl des Gastsystems führt zu einem Sprung in den Hypervisor
* dort wird geprüft, woher der Befehl kommt
* vom Gastsystem: Hypervisor veranlasst die Ausführung auf der Hardware im Namen des Gastsystems
* von einer Applikation: Hypervisor emuliert die Aktivitäten der Hardware
* 🡪 weniger Overhead