

Disclaimer

Die Übungen die hier gezeigt werden stammen aus der Vorlesung *Advanced Operating Systems*! Für die Korrektheit der Lösungen wird keine Gewähr gegeben.

1. Funktionale und nichtfunktionale Eigenschaften

- (a) Was ist eine nichtfunktionale Eigenschaft? Finden Sie Beispiele für sowohl funktionale als auch nichtfunktionale Eigenschaften
- eines Flugzeugs,
 - eines Smartphones,
 - eines Betriebssystems.

Solution:

- Flugzeugs
 - F: fliegen, bremsen,
 - NF: Innentemperatur halten, automatisierte Steuerung
- Smartphones
 - F: telefonieren ermöglichen, Internetzugang
 - NF: klein, leicht, energiesparend, strahlungsarm, umweltfreundlich,
- Betriebssystems
 - F: den Zugriff auf Daten ermöglichen
 - NF: leicht zu bedienen, skalierbar, offen, performant,

- (b) Charakterisieren Sie den Unterschied zwischen Laufzeiteigenschaften und Evolutionseigenschaften anhand je einer beispielhaften Eigenschaft.

Solution: Laufzeiteigenschaften Verfügbarkeit: während das System aktiv ist (läuft) muss es verfügbar sein. Überwachbar für ein einzelnes System nur über kurze Zeit. Evolutionseigenschaft Wartbarkeit: das System kann über eine längere Zeit überarbeitet/verbessert werden auch ohne dass dieses aktiv ist oder laufen muss. Überwachbar über eine Reihe von Systemen und nur eine lange Zeit

2. Anforderungen an Betriebssysteme

- (a) Welche grundlegenden funktionalen Eigenschaften muss jedes Betriebssystem qua definitione besitzen?

Solution:

- Hardware Multiplexen
- Hardware Schutz
- Hardware Abstraktion

- (b) Die Erfüllung spezieller nichtfunktionaler Eigenschaften kann Auswirkungen auf den gesamten Hard- und Softwarestack eines IT-Systems haben. Kennen Sie Anwendungsfälle für Betriebssysteme, die

- keinen Scheduler,
- kein Paging,
- keinen privilegierten Prozessormodus

benutzen? Recherchieren und begründen Sie, warum solche Designs möglich und sinnvoll sein können!

Solution: Embedded Systems mit nur einem Programm verwenden kein Scheduler oder Paging

3. Sparsamkeitsbegriff

- (a) Erläutern Sie die Begriffe „Sparsamkeit“ und „Effizienz“. Nennen und begründen Sie in diesem Zusammenhang mögliche Ziele eines sparsamen Betriebssystems.

Solution:

- (b) Ist Sparsamkeit grundsätzlich als Laufzeit- oder als Evolutionseigenschaft anzusehen? Begründen Sie anhand Ihrer Antwort auf Frage a).

Solution:

- (c) Begründen Sie anhand selbstgewählter Anwendungsszenarien, wann Effizienz im Umgang

- mit Energie
- mit Speicherplatz

eine zentrale nichtfunktionale Eigenschaft eines Betriebssystems ist.

Solution:

- (d) Sparsamkeit ist auch eine mögliche Anforderung an den Betrieb eines IT-Systems. Können Sie sich vorstellen, warum dabei „Sparsamkeit mit Funktionalität“ oder „Sparsamkeit mit Code“ eine Rolle spielen könnte? Bewerten Sie den aktuellen Linux-Mainline Kernel nach diesen Kriterien. . .

Solution:

4. Energieeffizienz

- (a) Welche hardwareseitigen Voraussetzungen müssen für energieeffizienten Betrieb eines Rechnersystems vorliegen? Welche Rolle spielt das Betriebssystem hierbei?

Solution: Hardware kann Energiesparmodi bereitstellen (sleep mode) und das Betriebssystem muss diese einstellen/umstellen können

- (b) Erklären Sie die Begriffe Reaktivität und Nutzererfahrung vor dem Hintergrund Ihrer Antwort auf Frage a).

Solution: Aus dem Energiesparenden Modi wird die Reaktivität niedriger sein aber kann durch das Betriebssystem für die jeweilige Anwendung angepasst wieder in einen Aktiven Modi versetzt werden. Z.B. Anpassung der Framerate einer GUI oder Grafik-intensiven Anwendung

5. Speichereffizienz

- (a) Erläutern Sie den Begriff Fragmentierung bei Realspeicherverwaltung. Warum kann dies auch für virtuelle Speicherverwaltung zum Problem werden?

Solution:

- (b) Welchen Einfluss hat die Seitentabelle für virtuelle Speicherverwaltung auf den unmittelbaren Speicherbedarf eines Betriebssystem-Kernels? Welchen haben etwaige Gerätetreiber? Und welche Möglichkeiten hat ein BS-Entwickler, mit beiden Problemen umzugehen?

Solution:

6. Betriebssystemarchitekturen für Sparsamkeit und Effizienz

Welche Vor- und Nachteile haben die beiden Architekturkonzepte Makro- und Mikrokernel für Entwurf und Implementierung energie- und speichereffizienter Betriebssysteme? Diskutieren Sie anhand der Betriebssysteme TinyOS und RIOT!

Solution:

7. Energieeffiziente Dateizugriffe Nehmen Sie an, eine Anwendung referenziert eine Folge von Festplattenblöcken: A, B, C, A, C, D, E, A . Zur Optimierung von sowohl Performanz als auch Energieverbrauch beim Zugriff auf diese Blöcke soll energieeffizientes Prefetching zum Einsatz kommen. Nehmen Sie weiterhin an, dass ein Blockzugriff (access) jeweils konstant 4 Zeiteinheiten (ZE) dauert, ein fetch oder prefetch je 1 ZE sowie dass der verwendete Festplattencache maximal 3 Blöcke fasst. Die gegebene Referenzfolge können Sie als gesichert annehmen (d. h. es sind keine Abweichungen im tatsächlichen Verhalten der Anwendung zu erwarten).

- (a) Entwickeln Sie zwei Zugriffsabläufe für diese Referenzfolge: einen, bei dem traditionelles Prefetching angewandt wird, sowie einen weiteren mit dem in Vorlesung besprochenen Verfahren zum energieeffizienten Prefetching.

Solution:

- (b) Begründen Sie anhand des Vergleichs beider Abläufe, welche Vorteile das energieeffiziente Verfahren in diesem Beispiel hat. Kann es Referenzfolgen geben, bei denen beide Verfahren (nahezu oder völlig) gleichen Energieverbrauch verursachen? Unter welchen Umständen, bzw. warum nicht?

Solution:

8. Energieeffizientes Scheduling

Schedulingalgorithmen, welche die optimale Reaktivität eines interaktiven Systems zum Ziel haben, basieren in der Regel auf Round Robin (RR) unter Einbeziehung von Prioritäten. Wenn zugleich Strategien zum sparsamen Umgang mit Prozessorenenergie durchgesetzt werden, können solche Algorithmen ihr Ziel jedoch verfehlen.

- (a) Begründen Sie obige These anhand einer geeignet konstruierten, beispielhaften Prozessmenge.

Solution:

- (b) Das Schedulingziel von RR („maximale Reaktivität“) lässt sich als die möglichst faire Verteilung von Rechenzeit auf Threads beschreiben. Finden Sie eine analoge Beschreibung des Schedulingziels „Energieeffizienz“ im Sinne der Antwort auf Frage a). Definieren Sie hierfür einen formalen Schedulingparameter analog zur Zeitscheibenlänge T bei RR.

Solution:

- (c) Implementieren Sie die einfache RR-Strategie (ohne Prioritäten) so, dass beide in Frage b) beschriebenen Schedulingziele verfolgt werden. Beachten Sie insbesondere, dass
- stark interaktive Threads durch verlängerte Antwortzeiten behindert sowie
 - energiesparende Threads durch relativ geringere Prozessorzeit benachteiligt werden können.

Solution:

Als Ablaufumgebung zur Simulation einer Prozessmenge können Sie entweder ein kleines multi-threaded Programm schreiben oder den Scheduling-Simulator PSSAV verwenden, den Sie evtl. noch aus der zweiten Übung im Grundlagenfach Betriebssysteme kennen. Im letzteren Fall können Sie bereits auf eine RR-Implementierung zurückgreifen; diese und die vom Simulator vorgesehenen Prozessinformationen müssen natürlich modifiziert werden. Stellen Sie Ihren Algorithmus (in knappem Pseudocode) im Seminar vor und zeigen Sie möglichst eine kleine Demonstration. Gehen Sie dabei auf die in der Vorlesung gestellten Implementierungsfragen ein. Tipps: Zur Simulation des Energieverbrauchs eines ablaufenden Prozesses dürfen Sie annehmen, dass dieser linear abhängig von der Prozesslaufzeit ist. Sie müssen daher dieses Prozessmerkmal lediglich als Faktor repräsentieren, der bspw. das verbleibende Energiebudget zum Zeitpunkt des Ablaufs einer Zeitscheibe errechnet oder bestimmt, wann ein zum Prozessstart gegebenes Energiebudget aufgebraucht sein wird (als zusätzlichen Schedulingzeitpunkt).