6 Hardwarevoraussetzungen (PageTable) 5 Punkte

In einem 32-Bit-System mit 4kB großen Seiten wird die zweistufige viruelle Adressierung eingesetzt.

a) Wie viele Speicher und wie viele Tabellen braucht man mindestens für die Speichertabellen(PD,PT) eines kleinen Programms(mit Code, Date, Heap und Stack)?

b) Für welche Bereiche des virtuellen Adressraums werden diese benötigt?

c) Wozu dient der TLB?

d) Wie wird die Virtuelle Adresse aufgeteilt (quantitativ)?

e) Wie "berechnet"die MMU dabei die reele Adresse (Zeichnung)?

f) Wie wird durch die virtuelle Adressierung verhindert, dass Programme auf fremden Adressraum zugreifen?

g) Was ist segmentierte Adressierung?

h) Wie wird aus der segmentierten Adresse die reele Adresse berechnet?

i) Wie wird aus der virtuellen Adresse die reelle Adresse?

a)
Bei 32-bit sind 12-bit offset und 20-bit Pagetable 2^{20} = 1048kB an Pagetablen also tabellen und 2^{12} OffsetAdressen. Somit wird durch ein kleines Programm mit zwei Stufiger Adressierung 1024kb*4kb= 4MB gebraucht. Dies ist viel zugroß dies könnte durch Page Direktorie kleiner gemacht werden, jedoch ist die Adresse zweistufig.
b)
keine Ahnung
c)
Die virtuelle Adressierung ist langsam da jedes mal zwei Zugriffe einmal auf die Page direkturie als auch auf die Page table getätigt werden muss. Weswegen Transmision lookaside Buffer genutzte wird dieser speicher die Reelen Seitenadresse und kann diese in einem Adresszyklus abrufen. Es ist ein voll assiozativer Cache.

f)
In der Page Table wird das W oder das P bit auf 0 gesetzt wenn das Programm damit arbeitet und damit der Schreibzugriff beschränkt
g)
Segmentierte Adressierung ermöglicht es Adressen einfach zu verschieben. Es hat eine Adressbreite von 20-bit ist aber nur mit 16-bit gefüllt dies 16-bit werden um 4-bit nach links geschoben und mit 0000 aufgefüllt. Dannach wird das Segmentregister darauf addiert Codesegment(CS), Stacksegment(SS), Heapsegment(HS), Datensegement(DS). Somit bleibt die grundlegende Adresse erhalten obwohl man diese verschiebt.

a) Was sind invertierte Seitentabellen?

b) Wie arbeitet diese Art der Adressierung?

a)+**b**)
Die invertierte Seitentabelle ist genau umgekehrt zur normalen jeder eintrag in der in der in vertierten Seitentabelle stellt ein Page da. Diese Page sucht nach dem Linken teil der virtuellen Adresse mit einer HashSuchalgoritmus bei Übereinstimmung wird der index der Seitentabelle als Seitennummer verwendet. i*4096 wird als anfangsadresse genutzt.

7 Dateisysteme 5 Punkte

[Hier ein Bild von einem ext4-Extent]

a) Beschreiben Sie die Funktionsweise des ext4-Dateisystems mit Extents(Bild)!

b) Warum kann ext4 mit Extents größerere Partitionen verwalten ohne Extents

c) Wie groß kann eine Datei werden (Blockgröße 4kB Rechenweg genügt)?

a)
Die Incode wird von 128 auf 254 erhöht um Zeitstempel zuermöglichen. Es sind variable Anzahl an Daten Blöcken die hintereinander im Speicher liegen. Es wird auf den ersten Verwiesen und die Anzahl der Blöcke mitgegeben. Statt den 15 bit im Incode gibt es einen Extens-Header und 4 Extens-Verweise.

b)
Wenn die 4 Extens verweise nicht ausreichen für die Datei können die 4 Extensverweise wiederrum auf eine weitere Liste mit 254 Extensverweise beinhaltet.
c)
Die Extens Blöcke können maximal 2^{15} bit groß werden 4kB=128Mb. Eine Datei kann maximal 2^{32} Blöcke groß werden. Gegeben sei ein Ext2-Dateisystem mit einer Blockgröße von 4 kB. Im Dateikopf seien 12 Einträge für die direkte Adressierung von Datenblöcken und 3 Einträge für die Verweise auf 1 bis 3-fach indizierte Blöcke.(Die Blocknumer sei 32 Bit groß)

a) Wie groß kann eine Datei werden (Blockgröße 4kB Rechenweg genügt)?

b) Wie kann man aus den Einträgen erkennen, in welchen Blöcken die Daten gespeichert sind(Zeichung)?

a)

$$12 \cdot (4kB/4B) + (4kB/4B) + (4kB/4B)^2 + (4kB/4B)^3 = 1TB$$

8 Prozesse (Threads)

a) Was ist ein Thread?

b) Wie verhalten sich Prozesse gegenüber Threads?

c) Was ist eine Race-Condition?

d) Wie kann man sie verhindern?

a) Was ist ein Prozess?

b) Welche Zustände kann er einnehmen?

c) Wie gelangt er von einem Zustand in einen anderen?

d) Wie ist sein Adressraum aufgebaut?

9 Fenstersystem

Beschreiben Sie die Funktionsweise eines Fenstersystems!

10 Prozesskommunikation

a) Vergleiche die Eigenschaften von Pipes und Message-Queues!

b) Warum kann man einzelne Signale nicht überladen?

c) Was ist ein Signal?

d) Wie werden sie behandelt?

e) Was ist ein Pipe? Welche Eigenschaften stellt sie bereit?

f) Was ist ein Semaphone

Ein Prozess möchte einem anderen Prozes Nachrichtenblöcke variabler Länge schicken. Der Empfänger möchte die Nachricht mit einem Aufruf lesen, ohne die Länge der Nachrichten zu kennen.

a) Welches Kommunikationsmittel sollte man dafür verwenden?

b) Wie kann der Empfänger vom Senden der Nachricht informiert werden, wenn er nicht dauernd nachfragen will?

a) Message Queues
b) Die Nachrichten werden im Message Queuen gespeichert und in der Reihenfolge des eintreffens Soriert aber der Empfänger kann auch nach einem Typ filtern der von der Message Queues zu gewissen wurde.

- Wie erkennt ein Betriebssystem, dass eine von einem Prozess benötigte Seite ausggelagert ist?
- Was tut sie es anschließend?

11 Systemaufrufe

Erklären Sie die Funktion und Wirkungsweise folgender System-Calls.

a) fork()

b) exec()

c) wait()

d) exit()

Warum wird bei Linux und Windows der Anwendungsadressraum von 4 GB auf 3 GB (Windows 2 GB) verkleinert? Welche Vorteile hat diese Maßnahme?

a) bei einem Systemcall

b) beim Datentransfer

b) Die schreib und Lese köpfe haben ein begrenzte Geschwindigkeit wenn man den Anwendungsadressraum verkleinert müssen die Schreib und leseköpfe sich weniger bewegen was den Prozess schneller macht.

- Warum führt copy_on_write zur beschleunigten Ausführung ein child-Prozesses nach fork()?
- Was geschieht, wenn einer der Prozesse einen Sprung in ein Unterprogramm ausführt?

12 Virtualisierung

Bei moderenen Systemen ist die Virtualisierung schon von Haus aus im Prozessor integriert.

a) Welchen zusätzlichen Hardwarebaustein braucht man um Nested Page Tables unterstützt wird?

b) Welche Eigenschaften muss ein Peripheriegerät vorweisen, um IO-Virtualisierung zu unterstützen?

a) Was ist ein Typ-1 und was ein Typ-2 Hypervisor?

b) Was ist die Voraussetzung ein Typ-1 Hypervisor?

13 Treiber

a) Was ist ein Device-Switch-Table?

b) Was bedeutet Minor- und Major-Number?

c) Was sind Block-Devices und Character-Devices?

d) Welche Aufgaben hat ein GErätetreiber?

e) Was versteht man unter "raw"und "cooked"?