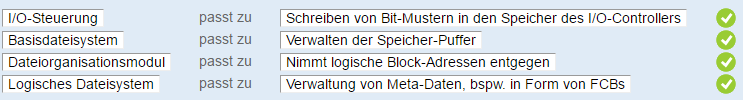
1. Ordne die nachfolgenden Aufgaben den entsprechenden Dateisystem-Ebenen zu.



1. Erkläre kurz die Funktion einer DMA im Vergleich mit konventionellem Speicherzugriff.

Konventionell werden Daten von einer E/A-Einheit in den Arbeitsspeicher über ein Prozessorregister als Zwischenspeicherung transferiert. Erst werden die Daten in das interne Register des Prozessors eingelesen, um dann im nächsten Schritt in den Arbeitsspeicher verschoben zu werden.

Der direkte Zugriff ist nun eine Schalt-/Steuermassnahme, die über Datenleitungen auf dem Motherboard eine Verbindungen zwischen Steckkarten und dem Arbeitsspeicher herstellt. So können die Daten ohne Umweg direkt in den Arbeitsspeicher geschrieben werden.

1. Welche Vorteile hat asynchrone Ein-/Ausgabe? Warum wird sie auch nicht-blockierend genannt?

Es muss zum Beispiel nicht darauf gewartet werden, dass sich der Scanner über das ganze Blatt bewegt hat. Bei synchroner E/A würde auf diese Beendigung gewartet und so das ganze Programm blockiert. Da bei asynchroner E/A nicht darauf gewartet werden muss, dass ein E/A-Vorgang abgeschlossen hat, blockiert es deshalb auch nicht.

1. Gewisse Geräte wie beispielsweise Scanner oder Tape-Drives können nicht auf sinnvolle Weise durch mehrere Applikationen gleichzeitig genutzt werden. Wie wird dieses Problem üblicherweise gelöst? Beschreibe die Funktionsweise und nenne den dazugehörigen Begriff.

Grundsätzlich über Buffer und Queues

Spooling: Zu bearbeitende Aufträge werden in einem Buffer gespeichert. Das Spooling System leitet dann die eingereihten (queued) 'spool data'  weiter.

Bsp Drucker: Die zu druckenden Dateien werden eingereiht. Das spooling system kopiert diese dann eine nach dem anderen zum Drucker, wenn dieser wieder soweit ist.

1. Beschreiben Sie wie DMA-basierte Ein-/Ausgabe funktioniert?

Die CPU startet den Transfer, dann macht sie andere Sachen während des Transports, wenn der Transport vollendet ist wird sie durch einen Interrupt vom DMA Controller benachrichtigt. So muss sie nicht warten, während der Transfer läuft.

1. Eine Freispeicher-Bitmap sehe nach der Formatierung wie folgt aus: 1000 0000 0000 0000

Wird Speicher belegt, so werde dieser jeweils an der am meisten links stehenden Stelle belegt. Wie sieht die Bitmap nach jedem der folgenden, in dieser Reihenfolge ausgeführten Schritte aus? Man nehme an, Dateien könnten z.B. durch einen Index auf mehrere frei Speicherbereiche verteilt werden.

(a) Datei A wird geschrieben und brauche 5 Blöcke  
(b) Datei B wird geschrieben und benutzt 8 Blöcke  
(c) Datei A wird gelöscht  
(d) Datei C wird geschrieben und benutze 7 Blocks

a) 1111 1100 0000 0000

b) 1111 1111 1111 1100

c) 1000 0011 1111 1100

d)1111 1111 1111 1111

1. Man nehme an, dass auf einer Disk 10 Datenblöcke 14-23 für die Speicherung von Datei-Inhalten reserviert seien. Es gebe zwei Dateien: f1 und f2. Der Start-Block von f1 sei bei 22, der Start-Block von f2 bei 16. Nachfolgend sind die Inhalte der FAT-Einträge gegeben. Welche Datenblöcke gehören f1, welche Datenblöcke gehören f2? Gib die jeweiligen Nummern an.

(14,18); (15,17); (16,23); (17,21); (18,20); (19,15); (20, −1); (21, −1); (22,19); (23,14)

Hinweis: (x,y) bedeutet: Block z besitzt einen Zeige zu Block y.

f1: 22 -> 19 -> 15 ->17 ->21

f2: 16 -> 23 -> 14 -> 18 ->20

1. Wieso ist eine System-Uhr weder Zeichen- noch Blockorientiert? Wie funktioniert der I/O einer Systemuhr?

Weil sie weder block-adressierbar ist, noch ein "character stream" produziert. Eine Systemuhr schickt lediglich Interrupts nach einem gewissen Intervall