20/20

Betriebssysteme

Übungsblatt 4

Micha Erkel Felix Ruh

7/7

Aufgabe 1

a) Der Prozessor läuft mit 800MHz = 800.000.000 $\frac{1}{s}$

Es wird mit einer Rate von $8\frac{MB}{sec} = 8 * 2^{20}\frac{Byte}{sec}$ gearbeitet.
Die Festplatte arbeitet allerdings nur 5% der Zeit, kann also auch nur 5% der Datenmenge in einer Sekunde übertragen.
-> $8 \cdot 2^{20}\frac{Byte}{sec} \cdot 5\% = 0,4 \cdot 2^{20}\frac{Byte}{sec}$

Wie viele 32-Bit (4-Byte) Wörter sind das?
$\frac{0,4 \cdot 2^{20}\frac{Byte}{sec}}{4Byte} = 0,1 \cdot 2^{20}\frac{Wörter}{sec}$

Davon werden immer 8 gleichzeitig versendet also:
$\frac{0,1 \cdot 2^{20}\frac{Wörter}{sec}}{8} = 0,0125 \cdot 2^{20}\frac{Übertragungen}{sec}$

Pro Übertragung (8 32-Bit-Wörtern) benötigt der Prozessor 1000 Zyklen.
-> $0,0125 \cdot 2^{20}\frac{Übertragungen}{sec} \cdot 1000 Zyklen = 12,5 \cdot 2^{20}\frac{Zyklen}{sec}$

Das sind in Prozent: $\frac{12,5 \cdot 2^{20}\frac{Zyklen}{sec}}{800.000.000\frac{1}{s}} \cdot 100 = 1,6384\%$

=> Der Prozessor muss also 1,64% seiner Rechenzeit für den Datenaustausch aufwenden.

b) Insgesamt benötigt der Prozessor für die Verarbeitung:
$1500 + 500 = 2000 \frac{zyklen}{16KB}$

Dabei gilt:
$16KB = 16 \cdot 2^{10} Byte$

Datenmenge pro Sekunde: $0,4 \cdot 2^{20} \frac{Byte}{sec}$

Wie viele 16-KB Pakete werden übertragen?
$\frac{0,4 \cdot 2^{20} \frac{Byte}{sec}}{16 \cdot 2^{10} Byte} = 25,6 \frac{Pakete}{s}$

Pro Übertragung benötigt der Prozessor 2000 Zyklen: $25,6 \frac{pakete}{s} \cdot 2.000 Zyklen = 51.200 \frac{Zyklen}{s}$

In Prozent:
$\frac{51.200 \frac{Zyklen}{s}}{800.000.000 \frac{1}{s}} \cdot 100 = 0.0064\%$

=> Das bedeutet, der Prozessor muss 0,0064% seiner Rechenzeit aufwenden.

Aufgabe 2

7/7

a) Bei Start des Systems wird /reset verwendet um den Zähler auf 0 zu initialisieren. Danach wird mit jedem Kommando auf INT der Zähler um 1 erhöht, bei jedem Kommando auf /INTA wieder gesenkt. Sodass immer die aktuelle Anzahl an Interrupts, welche der Prozessor verarbeiten muss, in dem Zähler gespeichert ist. Um festzustellen ob eine ISR auf dem Prozessor läuft, muss der Controller also lediglich den Zähler mit 0 vergleichen.

b) Nein reicht nicht.
Falls ein weitere Interrupt ankommt, weiß der Controller nicht welche Priorität dieser hat. Er stellt also entweder (falls der neue eine niedrigere Prio hat als der, der bereits bearbeitet wird) einen falschen Interrupt an die CPU durch oder (falls der neue eine höhere Pro hat) einen wichtigeren nicht durch.

Die Lösung:
Interrupt wird an Controller überstellt...

Prüfe, ob Zähler = 0.

1) Falls True stelle den Interrupt an die CPU durch, speichere den Interrupt in IVN - dessen Priorität in PR und erhöhe Zähler um 1.
2) Falls False vergleiche Priorität des neuen Interrupts mit der in PR gespeicherten.
    i. Falls die Priorität des neuen Interrupts höher ist, als die des aktuellen Interrupts. Verschiebe den aktuellen (im IVN) in den Stack an die Stelle <PR>. Übertrage den neuen Interrupt in IVN, dessen Priorität in PR, stelle ihn durch an die CPU und erhöhe den Zähler um 1.

wozu durchgehen, der Zähler zeigt doch auf die richtige Stelle und oben ist immer der mit der momentan höchsten Priorität?
ist nicht falsch. Ihr könntet aber den Zähler als Stackpointer mitbenutzen.

nicht notwendig darüber nachzudenken und es kommt auch auf die Implementierung an

ii. Andernfalls speicher den neuen Interrupt im Stack an der Stelle seiner Priorität. Erhöhe den Zähler um 1.

Nach Abarbeitung des, gerade in der CPU befindlichen Interrupts, sendet diese das /INTA Signal. Senke den Zähler um 1. Gehe den Stack von der aktuellen Priorität nach unten durch (<PR> nach 1), lade den ersten gefunden Interrupt in den IVN und seine Priorität in den PR. Setze seine Speicherzelle gleich 0 und stelle ihn durch an die CPU durch. Falls kein Interrupt gefunden werden konnte, kann der Controller auf den nächsten Interrupt von Außen warten.

Aufgabe 3

6/6