IBN - Blatt01

Luis Schulte, Gregor Nageler, Till Rostalski

04 Mai 2021

Link zum GitHub

Aufgabe 1

	Till	Gregor	URZ Heidelberg
Produkt	Macbook Air	HP Envy x360	IBM 360-44
Kosten	1300€	900€	3Mio DM
Taktfrequenz	1,6 GHz	2GHz	
Hauptspeicher	8 GB	8GB	256KB
nichtvolatiler Speicher	512 GB	500GB SSD	1,1MB
Betriebssystem	MacOS	W10, Ubuntu	IBSYS

Aufgabe 2

GPR: 16 x 64-bit registers $\rightarrow$ 1024 bit = 128 Byte
XMM: 16 x 128-bit registers $\rightarrow$ 2048 bit = 256 Byte
MMX: 8 x 64-bit registers $\rightarrow$ 512 bit = 64 Byte
Gesamt: 448 Byte, entspricht $4,17 * 10^{-5}$ % des Hauptspeichers.

Aufgabe 3

a. Umgebungsvariablen sind Variablen, die dem BS abhängig von dem Benutzer zur Verfügung stehen, um eine Arbeitsumgebung zu definieren. In diesen werden veränderbare Informationen gespeichert, die das BS, Prozesse o. Ä. für ihre Arbeit benötigen. Mit `printenv` kann man eine Liste

der Umgebungsvariablen des Nutzers ausgeben. Ein Beispiel ist die `HOME`-Variable, die den Pfad des Hauptverzeichnisses eines Benutzers speichert (in einem unixoiden BS bspw. `/Users/maxmustermann`).

b. Lesen von Umgebungsvariablen in Bash: `echo $VAR`

c. Schreiben von Umgebungsvariablen in Bash: `export KEY="value"` (nicht permanent; muss mit `source` aufgerufen werden; permanente Variablen können in `.bashrc` gesetzt werden)

d. Lesen von Umgebungsvariablen in C: `char* getenv(const char* name);`

e. Schreiben von Umgebungsvariablen in C:

```
int setenv(const char *name, const char *value, int overwrite);
```

Aufgabe 4

a. `man` oder `cat` oder `nl` oder `less`

b. Öffnen des Buchs `cat filename.txt &`

das `&` öffnet den Prozess im Hintergrund

Öffnen der Mail: `sudo less /var/mail/username &`

switchen zwischen den beiden Prozessen: `fg %[processID]` mit processID z.B. 1,2

c. Alternativ Kopieren `cp sourecefile destinationfile &`

switchen mit `fg %ID`

Es laufen dann mindestens drei Prozesse, der der Shell, der fürs Kopieren und der fürs Buch

Mit `[code] &` kann ein Prozess ausgeführt werden, der nicht die Eingabe auf der Shell blockiert, sondern im "Hintergrund weiterläuft". Dann laufen mindestens der Prozess, der die besagte Datei kopiert, und der, der das Lesen seines "Buchesßteuert.

Aufgabe 5

a. siehe Code

b. wir erwarten, dass alle Elemente im Array `thread_args` den Wert 0 haben. Wir überprüfen dies, indem wir am Ende der Main Funktion über den Array iterieren und dies asserten.

c. auf HP Envy x360 mit Ubuntu failed die assertion 0-5 mal. Auf MacBook Air mit MacOS failed die assertion ca. 20 mal.

Aufgabe 6

a. Parameter: lower und upper sind vom Typ int64 (Werte: 2, 10000) n_parts ist vom Typ int(Wert 8).

   let Variable step ist vom Typ int64(Wert: 10000-2 = 9998 /8 = 1249 (abgerundet)).

   let Variable lower_ends ist vom Typ eine Sequenz mit den Werten 2, 1251, 2500, 3749, 4998, 6247, 7496, 8745, 9994

   Variable upper_ends ist vom Typ eine Sequenz, sie hat die Werte 1251, 2500, 3749, 4998, 6247, 7496, 8745, 10000

   result ist vom Typ eine Seqeunz, sie hat die Werte (2,1251), (1251,2500), (2500,3749), (3749,4998), (4998,6247), (6247,7496), (7496,8745), (8745,10000)

b. proc deklariert eine procedure, der Name der Procedure ist subdivide_range, mit 3 Parametern (Typen siehe a.). Keyword auto bestimmt den Typ des Rückgabewertes.

   let deklariert eine Variable, welche nur einmal assigned werden kann und danach nicht mehr verändert wird. Mit upper-lower wird der Bereich bestimmt, der aufgeteilt werden muss. Durch Division mit div wird bestimmt, wie groß ein Part sein darf.

   In der Zeile mit lower_ends passiert folgendes: countup zählt von lower bis upper in step Schritten hoch. toSeq() erzeugt aus den von countup erzeugten Werten eine Sequenz. Die Sequenz wird in lower_ends gespeichert.

   In der nächsten Zeile wird upper_ends als var deklariert. Dies bedeutet, dass die Variable neu assigned werden kann. lower_ends wird automatisch zum Typ Sequenz. Ihr werden die Werte der Sequenz lower_ends von Index 1 bis zum vorletzen Index zugewiesen.

   Das Element upper wird an die Sequence upper_ends angefügt.

   Die Procedure zip() vereint die als Parameter übergebenen Sequences zu einer neuen Sequence, bestehend aus Tupeln der Elemente an den jeweils gleichen Indizes. Die Sequence wird in der Variable result"gespeichert.

Aufgabe 7

- **Focused mode** – Intensive Konzentration auf das Lernen oder Verstehen von etwas, das mittels bekannter Denkmuster erlernt werden kann. Man befindet sich, im Unterschied zur weniger konzentrierten Grundhaltung, in der man durch Ablenkung geplagt wird, im »Tunnel«. Bereits bekannte Denkmuster können angewandt werden – neue zu erlernen ist effektiver im Diffused Mode.

- **Diffused mode** – Breite Sicht (Überblick) auf das Spektrum (im Gegensatz zur tiefen (Ein-)sicht im Focused Mode). Neue Denkmuster können schneller erlernt, etabliert und angewandt werden.

a. **Vokabeln lernen** – (Abhängig von der angewandten Methode des Vokabellernens:) Für monotones und stumpfes Auswendiglernen der Definitionen von Vokablen genügt der Focused Mode. Die tiefe Konzentration, gepaart mit Wiederholung erlauben effektives Auswendiglernen. Sollen die Vokabeln in Verbindung mit weiteren Eigenschaften (bspw. Etymologie, Kontext der Kultur u. Ä.) des Wortes wirklich gelernt werden, dann ist der Diffused Mode gefragt – resp. eine Kombination.

b. **Konzepte aneignen** – Diffused Mode. Die Eigenschaften des Focused Mode, die tiefe Sicht (aber nicht breite!) auf die Dinge und das Anwenden bereits bekannter Konzepte und Denkmuster (aber nicht neuer!) im Focused Mode, erfüllen nicht die Notwendigkeiten für das Aneignen neuer Konzepte: breite Sicht, ein Konzept überblicken, nicht von dem bereit Bekannten behindert werden usw.

c. **Art einer Rechenaufgabe trainieren** – Focused Mode. Da nur eine spezielle Rechenaufgabe vertieft werden soll, ist der Focused Mode mit seiner intensiven Konzentration für bereits bekannte Denkmuster prädestiniert.

d. **Für IBN-Klausur lernen** – Das Vorbereiten auf eine Klausur erfordert aufgrund der mannigfachen Bereiche (konzeptuelles Wissen, Terminologie, Faktenwissen u.a.m.) die Kombination der Modi.

Salvador Dalí und Thomas Edison verfolgten eine interessante Strategie: Beide ließen sich in einem bequemen Sessel nieder, entspannten und verfolgten ihre Gedanken, so wie sie ihnen gerade erschienen. Der Fokus galt keinem bestimmten Objekt. Mit der Zeit verfielen beide in einen immer ruhigeren und entspannteren Zustand – bis das Einschlafen unausweichlich war. Einen Gegenstand wie einen kleinen Schlüsselbund, den sie sich vorher in die Hand

legten und der laut genug sein würde, um bei einem Aufprall mit dem Boden nach dem Loslassen der erschlafften Hand ein Geräusch zu erzeugen, benutzten sie gezielt, um einen Zustandswechsel der Modi zu forcieren: Der kreative Diffused Mode, in dem neue grobe Ideen gesammelt werden können, der während des Übergangs zum Schlafen eintritt, wird durch das plötzliche Aufwachen beendet. Der folgende Focused Mode erlaubt ein Vertiefen in die soeben gesammelten Anreize. Das Wechseln zwischen den Modi ist besonders effektiv für das Entwickeln und Ausarbeiten komplexer Ideen.