

Bitte bearbeiten Sie alle Aufgaben im Umfang von insgesamt **70 Punkten** und tragen Sie Ihre Antworten direkt in die Aufgabenblätter ein. Nutzen Sie in dieser Probeklausur auch die Vorlesungsfolien (in der Prüfung geht das nicht).

1. XML und XHTML (5 P.)

Bei der Überprüfung von XML-Dokumenten mit einer **Document Type Definition** (DTD) unterscheiden wir zwischen **wohlgeformten** und **validen** Dokumenten.

- a) Erklären Sie den Unterschied zwischen Wohlgeformtheit und Validität.
- b) XHTML ist eine XML-Sprache, die HTML nachbildet. Nennen Sie zwei Punkte, auf die Programmierer / Web-Entwickler bei XHTML achten müssen, während sie bei HTML keine Rolle spielen.

2. Zeichen-Kodierung (7 P.)

Das Textsatz-Programm LaTeX unterstützt aus historischen Gründen eine eigenwillige Kodierung für deutsche Umlaute: `\ "a` interpretiert das Programm als `ä`, entsprechend `\ "A` als `Ä`, `\ "o` als `ö` usw.; für das scharfe `s` (β) gibt es die Kurzform `\ "s`. (Der Backslash `\` ist in LaTeX das allgemeine Zeichen dafür, dass ein Steuerbefehl folgt. Dadurch ist es recht umständlich, einen echten Backslash in den Text zu bekommen: Es gibt dafür den Befehl `\textbackslash`.)

Bindet man ein spezielles Paket ein, das einige deutsche Anpassungen vornimmt, kann man bei den Umlauten jeweils auf den führenden Backslash verzichten; statt `\ "a` reicht also z. B. `"a`.

- a) Zum Starten: Schreiben Sie die folgenden Worte in beiden LaTeX-Notationen auf:

	LaTeX (mit Backslashes)	LaTeX (ohne Backslashes)
müßig		
Eßer :)		
Oeko & Öko		
Mämümö		

- b) Nutzt man die Kurzvarianten, gibt es ein generelles Problem mit Anführungszeichen – was für ein Problem könnte das sein?

c) Die „Herleitung“ dieser LaTeX-Kommandos ist so zu erklären:

\ = Achtung, es folgt ein Befehl

" = Der folgende Buchstabe bekommt ü-Punkte (passt beim „ß“ nicht ganz...)

u = es ist ein u mit ü-Punkten (also ein ü)

Raten Sie, wie die entsprechenden LaTeX-Kommandos aussehen, mit denen sich französische Sonderzeichen (é, è, ê etc.) erzeugen lassen:

é =

è =

ê =

d) Umlaute werden manchmal als Ä = Ae, ä=ae, Ö=Oe etc. geschrieben, wenn ein Ausgabegerät keine Umlaute beherrscht. Das kann zu unerwarteten Ergebnissen führen. Geben Sie ein Beispiel für ein Wort (und dessen spezielle Schreibweise), bei dem das Ergebnis dieser Umwandlung nicht befriedigend ist.

3. Archivformate (Windows und Linux)

(6 P.)

Windows und Linux verwenden unterschiedliche **Archivformate**, um Dateien zu komprimieren und / oder zu bündeln.

a) Ordnen Sie die folgenden Features den Archivformaten **tar** (Linux), **gzip** (Linux) und **zip** (Windows) zu.

tar	gzip	zip	
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Inhalte sind immer komprimiert
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Archivdatei enthält genau eine Originaldatei
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Archivdatei kann ganze Verzeichnisse enthalten
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Inhalte sind nie komprimiert

b) Betriebssystem-übergreifende Paketformate müssen Zusatzinformationen (neben Dateiname und Dateieinhalt) speichern. Nennen Sie zwei Arten von Zusatzinformationen (und jeweils das zugehörige Betriebssystem), die beim Entpacken unter einem anderen Betriebssystem Probleme verursachen können.

4. Software-Ergonomie

(6 P.)

- a) Kriterien für Benutzerfreundlichkeit und Gebrauchstauglichkeit von Software sind u. a. **Selbstbeschreibungsfähigkeit** und **Lernförderlichkeit**. Nennen und erklären Sie für alle drei Kriterien jeweils ein Beispielverhalten oder -Feature eines Programms, das dieses Kriterium erfüllt.

Selbstbeschreibungsfähigkeit:

Lernförderlichkeit:

- b) Nennen Sie (ohne Erklärung) zwei weitere Kriterien aus der Liste der (insgesamt sieben) Kriterien.

5. Von-Neumann-Architektur

(7 P.)

- a) Der Universalrechner (von-Neumann-Rechner) besteht aus fünf Komponenten. Nennen Sie diese fünf Komponenten. (1 P)

- b) Für ein Programm müssen der Programmcode und die Programmdateien im Speicher vorgehalten werden. Verwenden von-Neumann-Rechner gemeinsame oder getrennte Speichersysteme für Code und Daten? (2 P)

Nennen Sie jeweils einen Vor- und einen Nachteil von getrennten Speichersystemen. (2 P)

- c) Was ist die „Pseudo-Harvard-Architektur“? (2 P)

6. RISC vs. CISC

(6 P.)

Im Folgenden finden Sie sechs fiktive Befehle, deren jeweilige Funktion neben dem Kommando beschrieben ist.

- | | |
|--------------------|---|
| 1) INCR2 Reg | erhöht den Wert im angegebenen Register um 2 |
| 2) INCR2 Mem | erhöht den Wert im Hauptspeicher (an der angeg. Adresse) um 2 |
| 3) SWAP Mem1,Mem2 | vertauscht die Inhalte der Speicherzellen Mem1, Mem2 |
| 4) CHECK Reg,o1,o2 | Prüft Inhalt des angegebenen Registers.
– Falls Inhalt < 0: relativer Sprung um o1 Byte vorwärts
– Falls Inhalt > 0: relativer Sprung um o2 Byte vorwärts
– Falls Inhalt = 0: fortfahren an nächster Adresse |
| 5) SAVEALL Mem | schreibt den Inhalt sämtlicher Register in den Hauptspeicher (ab Adresse Mem) |
| 6) LDBLK Reg,Mem | liest einen durch Reg festgelegten 1 KB großen Block von der Festplatte und speichert ihn im Speicher ab Adresse Adr |

Kreuzen Sie in der folgenden Tabelle an, welche dieser Befehle es auf einer RISC- bzw. CISC-Architektur geben kann. (Es muss keine konkrete CPU geben, die den jeweiligen Befehl bietet.)

	RISC	CISC
1) INCR2 Reg		
2) INCR2 Mem		
3) SWAP Mem1,Mem2		
4) CHECK Reg,o1,o2		
5) SAVEALL Mem		
6) LDBLK Reg,Mem		

7. Abhängigkeit

(7 P.)

Zeichnen Sie für folgendes Programm den Abhängigkeitsgraph (ohne transitive Pfeile, ohne RAR-Abhängigkeiten). (5 P)

```

1   a   IS   $0
2   b   IS   $1
3   t   IS   $2
4   y   IS   $3

5           MUL  t, a, a
6           SUB  t, t, b
7           SUB  t, t, b
8           MUL  y, b, b
9           SUB  y, y, t

```

Geben Sie außerdem an, welche Funktion das Programm berechnet, wenn a und b Argumente, y das Ergebnis und t ein Zwischenwert sind. (2 P)

$$y = f(a,b) =$$

8. Superskalarität

(8 P.)

- a) Was bedeutet „superskalar“ im Vergleich zu „skalar“? (2 P)
- b) Eine CPU-Pipeline bestehe aus 100 Stufen. Fast alle Stufen benötigen zum Erledigen ihrer jeweiligen Aufgabe ca. 1 ns, aber drei der Stufen benötigen ca. 4 ns.
- (i) Warum ist das kein vernünftiges Design? (2 P)
 - (ii) Wie lässt sich relativ leicht ein vernünftiges Design aus der bestehenden 100-Stufen-Pipeline entwickeln? (2 P)
- c) In der 5-stufigen RISC-Pipeline kann es zu Kollisionen kommen. Was ist eine Kollision und welche Kollisionen sind hier möglich? (2 P)

9. Stack-Maschine

(5 P.)

Schreiben Sie ein Programm für die Stack-Maschine, das die folgende Funktion f berechnet:

$$f(a,b,c) = (a^2+b^2+c^2) / (a+b+c)$$

Dabei können Sie davon ausgehen, dass a , b , c bereits auf dem Stack liegen (der erste POP-Aufruf holt c vom Stack). Sie können Speicheradressen ab $0x1000$ frei zum Speichern von Zwischenergebnissen verwenden. Das Endergebnis soll nach Ende Ihres Codes auf dem Stack liegen.

10. Reorder Buffer

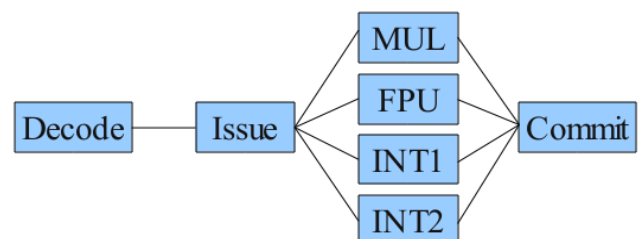
(9 P.)

Betrachten Sie die folgenden vier Befehle:

- 1 FADD \$5, \$1, \$2
- 2 FSUB \$6, \$1, \$2
- 3 FMUL \$1, \$5, \$6
- 4 FADD \$1, \$1, 1

Die Befehle FMUL, FADD und FSUB benötigen jeweils vier Zyklen auf der MUL- bzw. FPU-Ausführungseinheit.

Zeigen Sie, wie diese Befehle auf einem System mit Reorder Buffer (zwei Zuteilungen pro Takt) ausgeführt werden; zeichnen Sie dazu die Inhalte des Reorder Buffers zu den 17 Zeitpunkten t_0 bis t_{16} ein.



0

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

1

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

2

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

3

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

4

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

5

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

6

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

7

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

8

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

9

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

10

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

11

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

12

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

13

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

14

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

15

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

16

			Operanden				
Befehl	unit	Status	Q_Y	Q_Z	V_Y	V_Z	Ergebnis

11. PC als Arbeitsplatz

(4 P.)

Welche der folgenden Aussagen sind korrekt? Bitte die richtigen Aussagen ankreuzen.

- Computer-Algebra-Systeme (CAS) haben besondere Stärken in der numerischen Mathematik: Sie berechnen Lösungen durch Näherungsverfahren.
- In der Versionsverwaltung entsteht ein „Branch“, wenn die aktuellste Version einer Datei bearbeitet und in veränderter Form eingecheckt wird.
- „Branches“ kann man durch eine „Merge“-Operation wieder zusammenführen.
- Beim Erstellen von LaTeX-Dokumenten konzentrieren Autoren sich in erster Linie auf den Inhalt und nicht auf die Form.