

## Prüfungsfragen Rechnerarchitektur 2007

Rechnerarchitektur (WS 2006/07),  
Design und Organisation einer Linux-Distribution (WS 2005/06)

Dr.-Ing. Volkmar Sieh, Beisitzer:  
Wolfgang Hohl

März 2007

### Bemerkungen zu Prüfung und Prüfer

- kein Schwerpunktfach
- Ergebnis: 1,0
  
- sehr lockere Atmosphäre, wie man es von der Vorlesung her auch vermuten würde
- lässt aussuchen, mit welchem Thema man anfangen will (ich habe RA gewählt)
- Gespräch lässt sich lenken, man kann viel erzählen, ohne unterbrochen zu werden
- Reaktion „jaaa“ (in Ja-Aber-Intonation) ist nicht negativ aufzufassen (kann bei mir oft, aber habe trotzdem die Bestnote erhalten), also nicht irritieren lassen
- teilweise helfen einem die empfohlenen Bücher beim Verständnis weiter (Überfliegen und genaueres Lesen bei den relevanten Stellen reicht)

### Fragen zu RA

- Ziele und Maße – was gibt es da so?  
⇒ erst Grobüberblick mit jeweils kurzer Erläuterung: Performance, Netzwerk, Leistungsaufnahme, Skalierbarkeit, Zuverlässigkeit
- Performance genauer – was hat darauf Einfluss?  
⇒ CPU, Busse, Speicher, I/O
- I/O – wie kriegt man das schnell hin?  
⇒ komplette Entwicklung mit jeweils Beschreibung: Polling, IRQ-Driven, Buffered, Independent, DMA, Intelligent I/O

DMA-Vorteile: 1 statt 2 Buszyklen, CPU kann im Cache weiterrechnen

Intelligent I/O mit Scatter (Beispiel Page auf HD-Blöcke) und Gather (NW-Paket aus Header und Payload) und Buffer- und Operation-Chaining

- I/O-Ziel wenige IRQs – warum?  
⇒ CPU sollte lieber rechnen, außerdem Pipelining und Superskalaritäts (SS)-Probleme
- Pipelining – Beschreibung und Probleme?  
⇒ Structural Hazards, dagegen z.B. Stalls oder Harvard-Architektur mit I- und D-Caches
- ⇒ Data Hazards, RAW: durch Stalls oder Forwarding, Überlappung der Stages erklären
- ⇒ Data Hazards, WAR/WAW: nur bei SS-Out-of-Order-Issuing, Lösung durch Renaming, Reorder-Buffer und Retire-Phase
- ⇒ Control Hazards: Delay Slots (wie viele (3/1 je nach Implementierung); Problem: nur schlecht füllbar) vs. Stalling/Abbrechen

Prediction-Arten: Wired, Static, Dynamic, Direction-Based

Dynamic: BTAC-Funktionsweise, Erweiterung zu BTB

### Fragen zu DOLINUX

- Pakete – wofür?  
⇒ Abhängigkeiten: erkennbar (Libraries, Interpreter) vs. nicht (System-Calls, z.B. Mutt braucht MTA)
- klein (komplexe Abhängigkeiten) vs. groß (Konflikte, viel Unnötiges)
- Gruppierung: Anwendung, Library, Tool-sammlungen, Development, Doku
- Abhängigkeiten – nicht nur zur Laufzeit, sondern auch?  
⇒ zur Build-Zeit: Libraries, Sourcen, Kernel-API – falsche Version, Fehlen, Fehler

Damit wir auch in Zukunft aktuelle Prüfungsfragen haben, sind wir auf Deine Mithilfe angewiesen. Bitte maile uns die Fragen Deiner Prüfung, ein Formular dazu findest Du auf unserer Homepage: <http://fsi.informatik.uni-erlangen.de/>

- SPEC-File-Aufbau erklären und welche Teile für Build und welche für Installation
  - Cross-Build-System: Cross-Compiler, Cross-Linker, eigene Libraries ⇒ damit Sourcen kompilieren
  - Wie bringt man das auf den Target-PC?
- ⇒ Booten kurz erklären
- ⇒ Kernel-Config (InitRD-FS einkompiliert) und InitRD, Module nachladen (IDE-Controller, ISOFS)
- ⇒ CD finden (proc-FS und Magic-Datei) und mounten, dann Basissystem kopieren und konfigurieren

Damit wir auch in Zukunft aktuelle Prüfungsfragen haben, sind wir auf Deine Mithilfe angewiesen. Bitte maile uns die Fragen Deiner Prüfung, ein Formular dazu findest Du auf unserer Homepage:  
<http://fsi.informatik.uni-erlangen.de/>