

Übung zu Betriebssysteme

Graphische Anwendung

30. & 31. Januar

Andreas Ziegler, Bernhard Heinloth, Christian Eichler

Lehrstuhl für Informatik 4
Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme



FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Eine (graphische) Anwendung

Freiwillige Aufgabe

Die Vorgabe enthält

- einen Zufallszahlengenerator
- ein Dateisystem
 - *Minix v3*, von Linux in PASST geklaut
 - mit typischen Schnittstellen
- einen Grafikmodus (VESA)
 - eine *PNG* Bibliothek
- ein kleines Beispiel

Freiwillige Aufgabe

Die Vorgabe enthält

- einen Zufallszahlengenerator
- ein Dateisystem
 - *Minix v3*, von Linux in PASST geklaut
 - mit typischen Schnittstellen
- einen Grafikmodus (VESA)
 - eine *PNG* Bibliothek
- ein kleines Beispiel

Ihr braucht noch

- eine dynamische Speicherverwaltung
(`malloc()` / `free()`, z.B. *Halde* aus SP)

Freiwillige Aufgabe

Die Vorgabe enthält

- einen Zufallszahlengenerator
- ein Dateisystem
 - *Minix v3*, von Linux in PASST geklaut
 - mit typischen Schnittstellen
- einen Grafikmodus (VESA)
 - eine *PNG* Bibliothek
- ein kleines Beispiel

Ihr braucht noch

- eine dynamische Speicherverwaltung
(`malloc()` / `free()`, z.B. *Halde* aus SP)

Macht was tolles daraus (wenn ihr wollt & Zeit habt).

Freiwillige Aufgabe

Die Vorgabe enthält

- einen Zufallszahlengenerator
- ein Dateisystem
 - *Minix v3*, von Linux in PASST geklaut
 - mit typischen Schnittstellen
- einen Grafikmodus (VESA)
 - eine *PNG* Bibliothek
- ein kleines Beispiel

Ihr braucht noch

- eine dynamische Speicherverwaltung
(`malloc()` / `free()`, z.B. *Halde* aus SP)

Macht was tolles daraus (wenn ihr wollt & Zeit habt).

Und schickt uns das Ergebnis.

Freiwillige Aufgabe

Die Vorgabe enthält

- einen Zufallszahlengenerator
- ein Dateisystem
 - *Minix v3*, von Linux in PASST geklaut
 - mit typischen Schnittstellen
- einen Grafikmodus (VESA)
 - eine *PNG* Bibliothek
- ein kleines Beispiel

Ihr braucht noch

- eine dynamische Speicherverwaltung
(`malloc()` / `free()`, z.B. *Halde* aus SP)

Macht was tolles daraus (wenn ihr wollt & Zeit habt).

Und schickt uns das Ergebnis. Irgendwann.

Ruhmeshalle

via Netboot Ruhmeshalle
oder QEMU/KVM über [/proj/i4bs/halloffame](#)

Ruhmeshalle

via Netboot Ruhmeshalle
oder QEMU/KVM über [/proj/i4bs/halloffame](#)
(und bald auch auf der Webseite)

Ruhmeshalle

via Netboot Ruhmeshalle
oder QEMU/KVM über `/proj/i4bs/halloffame`
(und bald auch auf der Webseite)



teilweise Probleme wegen zu neuer Hardware

Prüfung

Prüfung (☠)

- Geprüft wird der Stoff der Vorlesung
 - ihr müsst **nicht** den Quellcode (auswendig) kennen
 - aber das Prinzip müsst ihr erklären können!
 - übt mit Kommilitonen, erklärt euch gegenseitig die Vorgehensweise
- alte Prüfungsprotokolle online bei der FSI Informatik:
<https://fsi.cs.fau.de/dw/pruefungen/hauptstudium/ls4/bs>
(Schreibt doch nach der Prüfung selbst eines!)

Judgement Day

- kommt [über]pünktlich
- und ausgeschlafen 😊
- ein Prüfer und ein protokollierender Beisitzer
- statt schweigend zu denken, lieber eure Überlegung aussprechen
→ wir können nur bepunkten, was ihr von euch gebt (und der Prüfer kann euch auf den richtigen Weg bringen)
- sollten Worte fehlen/nicht ausreichen, so habt ihr Stift und Papier
- die 30 Minuten sind schnell vorbei
- ihr bekommt nach weiteren 1-5 Minuten eure Note

Ausblick

Aufgaben

1. User mode (Ring 3)
2. System calls
3. Paging
4. Separation of kernel and user code
5. Fork
6. Message passing
7. Extended paging (copy-on-write)

Evaluation

- **36** aktive Übungsteilnehmer

- **36** aktive Übungsteilnehmer
- **30** TANs ausgeben

- **36** aktive Übungsteilnehmer
- **30** TANS ausgeben
- **22** Rückläufer

- **36** aktive Übungsteilnehmer
- **30** TANS ausgeben
- **22** Rückläufer → **73%**

**Danke für das Semester
&
vielleicht sieht man sich in BST!**

Viel Erfolg bei der Prüfung!