Übung zu Betriebssysteme

Aufgabe 7: Eine (graphische) Anwendung, Prüfung, Evaluation & Ausblick

7. Februar 2024

Maximilian Ott, Dustin Nguyen, Phillip Raffeck & Bernhard Heinloth

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

Lehrstuhl für Verteilte Systeme
und Betriebssysteme

Friedrich-Alexander-Universität
Technische Fakultät

Ruhmeshalle

via Netboot `Ruhmeshalle`
oder QEMU/KVM über `/proj/i4bs/halloffame`

Ruhmeshalle

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

Ruhmeshalle

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

teilweise Probleme wegen zu neuer Hardware

Anwendung für StuBS *(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

Anwendung für StuBS *(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)

Anwendung für StuBS *(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).

Anwendung für StuBS *(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.

Anwendung für StuBS *(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.

Prüfung

Prüfung (☠)

Vorbereitung

- 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

Vorbereitung

- 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!)

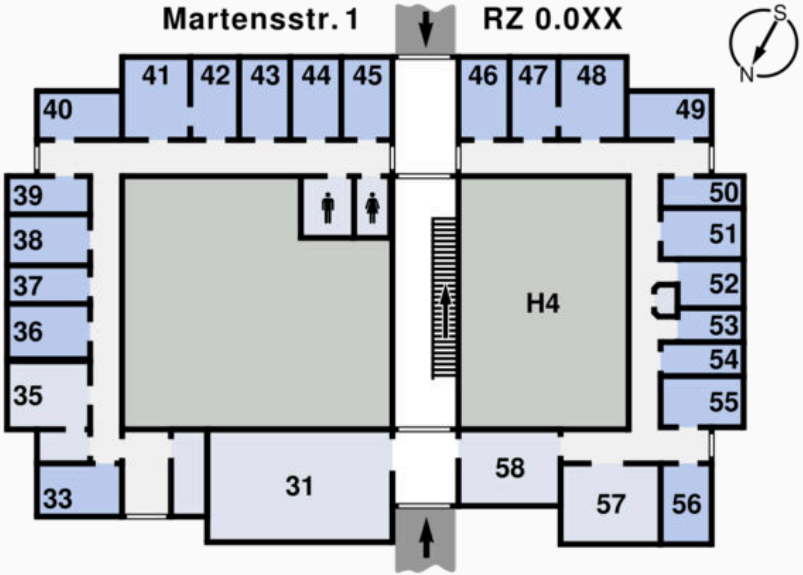
Vorbereitung

- 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!)
- *bei Prüfungsabsage:* Bitte immer eine (kurze) Mail
  - immer. Egal wie kurzfristig.
  - aber je früher desto besser
  - ggf. auch gleich Wunschersatztermin

Judgement Day: Präsenz

- kommt [über]pünktlich
- und ausgeschlafen ☺
- ein Prüfer und ein protokollierender Beisitzer
- statt schweigend zu denken, lieber eure Überlegung aussprechen
  - ➔ man darf 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
- 0.035 (0.049 als Ausweichbüro)

Prüfungsraum



Ausblick

BST Vorlesung

- Einflussfaktoren
  I. Systemaufruf
  II. Betriebssystemarchitektur
  III. Hierarchien
- Adressraumkonzepte
  I. Seiten: ein-/mehrstufig, invertiert, überwacht
  II. Segmentierung; seitenbasierte Hybride
- Adressraummodelle
  I. Mehradressraumsystem; total/partiell privat
  II. Einadressraumsystem
- Spezialfälle
  I. Dynamisches Binden (Multics)
  II. Übersetzungspuffer, adaptiver Speicherschutz (HW/SW; Kohärenz (IPI))
  III. Virtuell gemeinsamer Speicher (PEACE, OctoPOS)
  IV. Virtuell nicht-flüchtiger Speicher (PAVE)

BST Übung

StuBSmI – Studentisches Betriebssystem mit Isolation

BST Übung

StuBSmI – Studentisches Betriebssystem mit Isolation

1. Privilegientrennung

BST Übung

StuBSmI – Studentisches Betriebssystem mit Isolation

1. Privilegientrennung
2. Systemaufrufschnittstelle

BST Übung

StuBSmI – Studentisches Betriebssystem mit Isolation

1. Privilegientrennung
2. Systemaufrufschnittstelle
3. Paging

BST Übung

StuBSmI – Studentisches Betriebssystem mit Isolation

1. Privilegientrennung
2. Systemaufrufschnittstelle
3. Paging
4. Externe Anwendungen

BST Übung

StuBSmI – Studentisches Betriebssystem mit Isolation

1. Privilegientrennung
2. Systemaufrufschnittstelle
3. Paging
4. Externe Anwendungen
5. Erweiterte Prozess- und Speicherverwaltung

BST Übung

StuBSmI – Studentisches Betriebssystem mit Isolation

1. Privilegientrennung
2. Systemaufrufschnittstelle
3. Paging
4. Externe Anwendungen
5. Erweiterte Prozess- und Speicherverwaltung
6. Nachrichtenaustausch

BST Übung

StuBSmI – Studentisches Betriebssystem mit Isolation

1. Privilegientrennung
2. Systemaufrufschnittstelle
3. Paging
4. Externe Anwendungen
5. Erweiterte Prozess- und Speicherverwaltung
6. Nachrichtenaustausch
7. Erweitertes Paging

Abschlussarbeiten

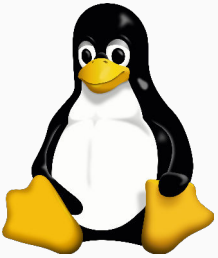
ODPOSs On-Demand Privileges for OSs

BA, MP, MA

Entwicklung eines alternativen Systemaufruf-Mechanismus für x86_64

- Ziel: Systemaufrufe ohne überflüssige Privilegienwechsel
- Idee: Privilegienwechsel bei Bedarf
  ⇒ ~~`int 0x80`~~, ~~`syscall`~~, ~~`sysenter`~~, `call` ✓
  ⇒ Behandlung des *General-Protection Faults*
- Proof-of-Concept Implementierung für arm64: ”Syscalls are dead, long live Syscalls”

  https://doi.org/10.25593/issn.2191-5008/CS-2023-03



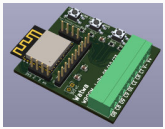
https://isc.tamu.edu/~lewing/linux

Bei Interesse: ott@cs.fau.de

watwa-OS

BA, MP, MA

Entwicklung eines minimalistischen
OS für energiegewahre, eingebettete Echtzeitsysteme



- Ziel: Energieeinsparungen, wo möglich
- klassische OS nicht geeignet:
  unbenötigte Gimmicks, viel Code, Laufzeitoverhead,
  keine Echtzeitgarantien

\+ Kontrolle über Clock-Distribution-Network:
CPU-Takt, Gerätesteuerung/konfiguration, ...

- viel Potential, wenn man die Kontrolle hat

  → https://cris.fau.de/converis/portal/publication/297648457

⇒ Entwicklung eines eigenen Betriebssystems

Bei Interesse: dengler@cs.fau.de, waegemann@cs.fau.de

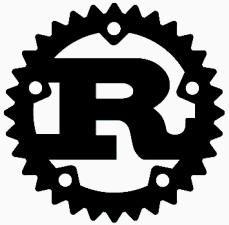
Partitioning Rust programs

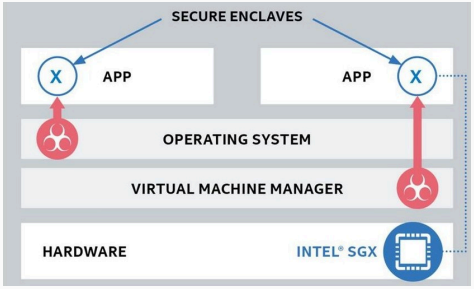
Wer am besten partitioniert, gewinnt

BA, MP, MA

Rust

- Speichersicherheit
- Statische Typisierung
- LLVM Frontend





SGX & Co.

- **Hardware** extension
- bietet starke **Sicherheitsgarantien**
- Kein **Vertrauen** ggü. Cloud Provider nötig

Ziel: Hardware optimal ausnutzen für maximale **Sicherheit**

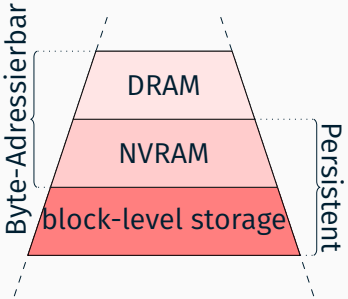
Tools: **Rust**, Statische Analyse, Symbolic Execution, & eigene Ideen?

Bei Interesse: onciul@cs.fau.de

Einschub: Nicht-flüchtiger Hauptspeicher

Eigenschaften von Intel® Optane™ Persistent Memory

- Byte-Adressierbar
- persistent
- Kapazität in TiB
- Schneller als Flash, langsamer als DRAM
- gutes Preis/Kapazitätz-Verhältnis



NEON (Linux + Optane)

BA, MP, MA

- Linux besitzt einige (periodisch ausgeführte) Persistenzmaßnahmen
- Aber: Unnötiger Overhead (Energie + Performanz) für den Betrieb auf Intel Optane (allgemein NVM)
- Idee: Identifikation + Entfernung vorhandener Persistenzmaßnahmen im Kernel

Bei Interesse: preisner@cs.fau.de

PAVE: Stromausfalltoleranz

BA, MP, MA

Übersicht

- NVRAM als Hauptspeicher
- DRAM als Cache
- Zustandssicherung bei Stromausfall

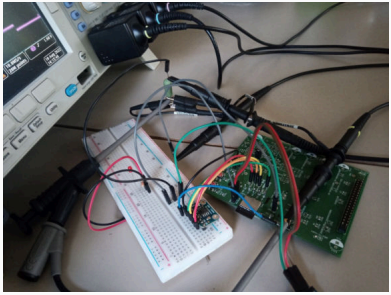
Themen

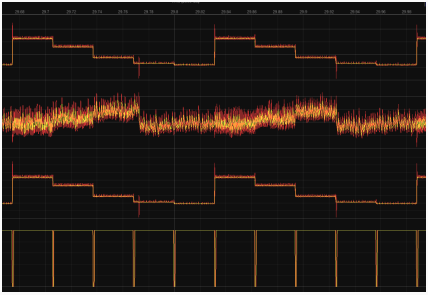
- Zugriffsmuster auf Seiten klassifizieren
- Rekonstruktion von Gerätezustand
- Identifikation rekonstruierbarer Strukturen
- Hypervisor-basiertes Tracing kritischer Abschnitte

Bei Interesse:
nguyen@cs.fau.de

Offene Stellen

EZS-Hiwi/Tutor





- HiWi/Tutor für Erweiterung von EZS gesucht
- neuer Fokus: Energiemessungen
- Bei Interesse: Meldet euch bei uns! Persönlich oder über `i4ezs-owner@lists.cs.fau.de`

Viel Erfolg bei der Prüfung
und schöne Semesterferien!