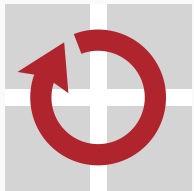
Honey, I shrunk the Snaps

Praktikum angewandte Systemsoftwaretechnik

13.03.2020

Michael Kupfer, Kay Friedrich

Friedrich-Alexander-Universität Erlangen-Nürnberg



Lehrstuhl für Verteilte Systeme
und Betriebssysteme

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Zur Erinnerung

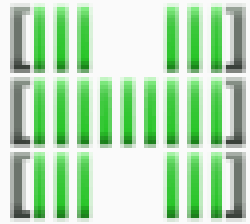
„Honey, I shrunk the ELFs”:

1. Statische Analyse
2. Dynamische Analyse
3. Überschreiben nicht genutzter Symbole
4. Löschen nicht genutzter Seiten

Ziele des Projekts:

- Anwenden der Toolchain auf Snap-Pakete
- Auswertung: Wie sehr lassen sich Snap-Pakete schrumpfen?
- Reduzierte Snap-Pakete neu verpacken

htop



- Nachbildung Programmstart:
  - Umgebungsvariablen setzen
  - Wrapper-Skripte
    $\Rightarrow$ `/bin/bash` als Entrypoint mit aufnehmen
- Debug-Symbole für Bibliotheken in Ubuntu-Archiven suchen
  $\Rightarrow$ web-crawler implementieren
- Snaps als `squashfs`
  $\Rightarrow$ Geringfügiges Abändern der Toolchain ohne Schreibzugriff

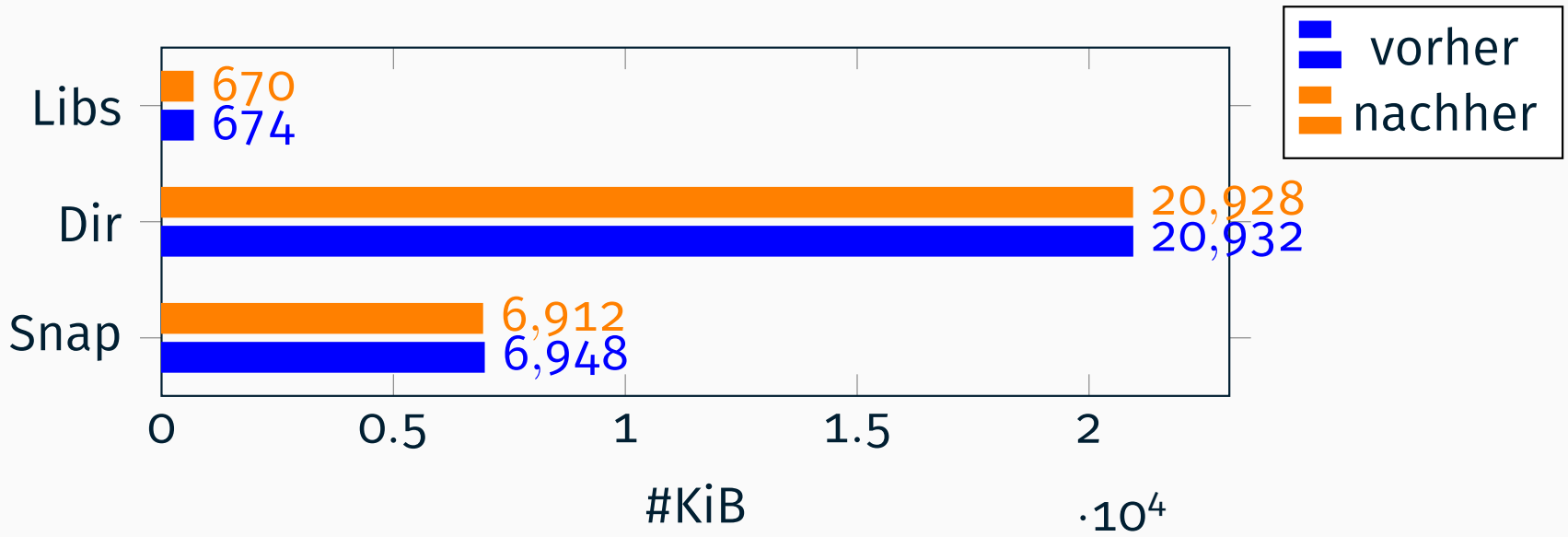
evince

- Änderungen in der uprobe-Implementierung seit erstellen der Toolchain
  $\rightarrow$ Keine uprobes mit mehr als 64 Zeichen möglich.
  $\Rightarrow$ Namensgebung für uprobes ändern

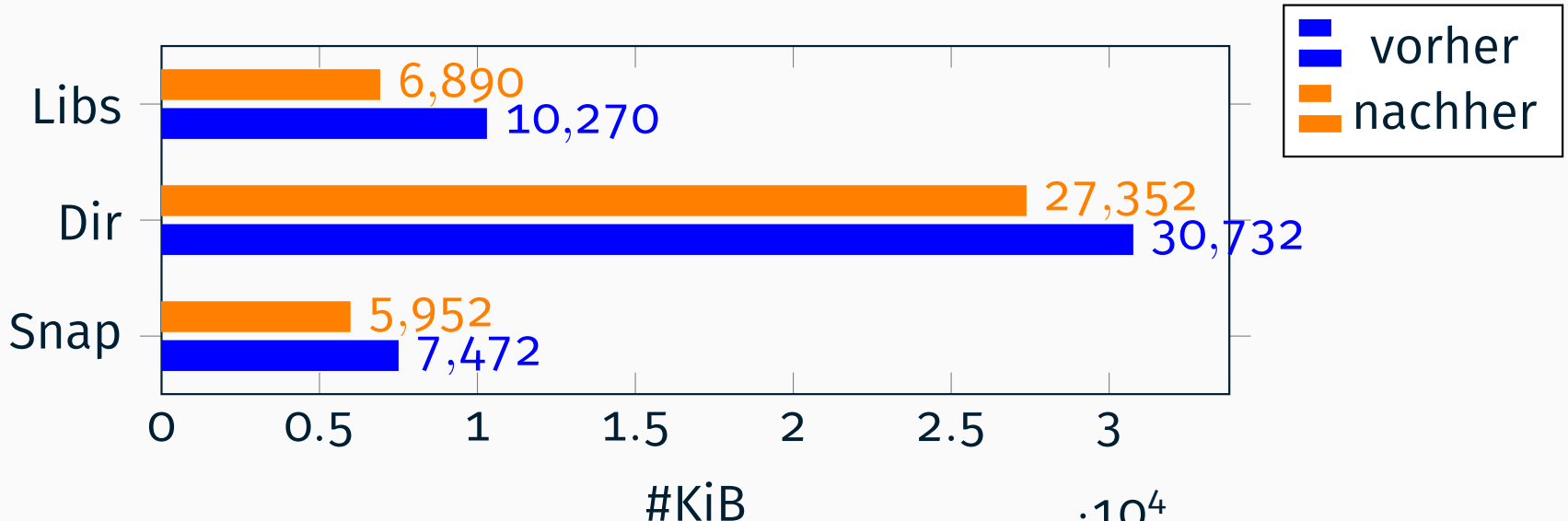
- Bibliotheken in Gnome werden teilweise über Umwege geladen
  $\rightarrow$`$SNAP/bin/evince` nicht ausreichend als Entrypoint der statischen Analyse
  $\Rightarrow$ Betroffene Bibliotheken als Entrypoints setzen

Statistiken - htop und evince

htop

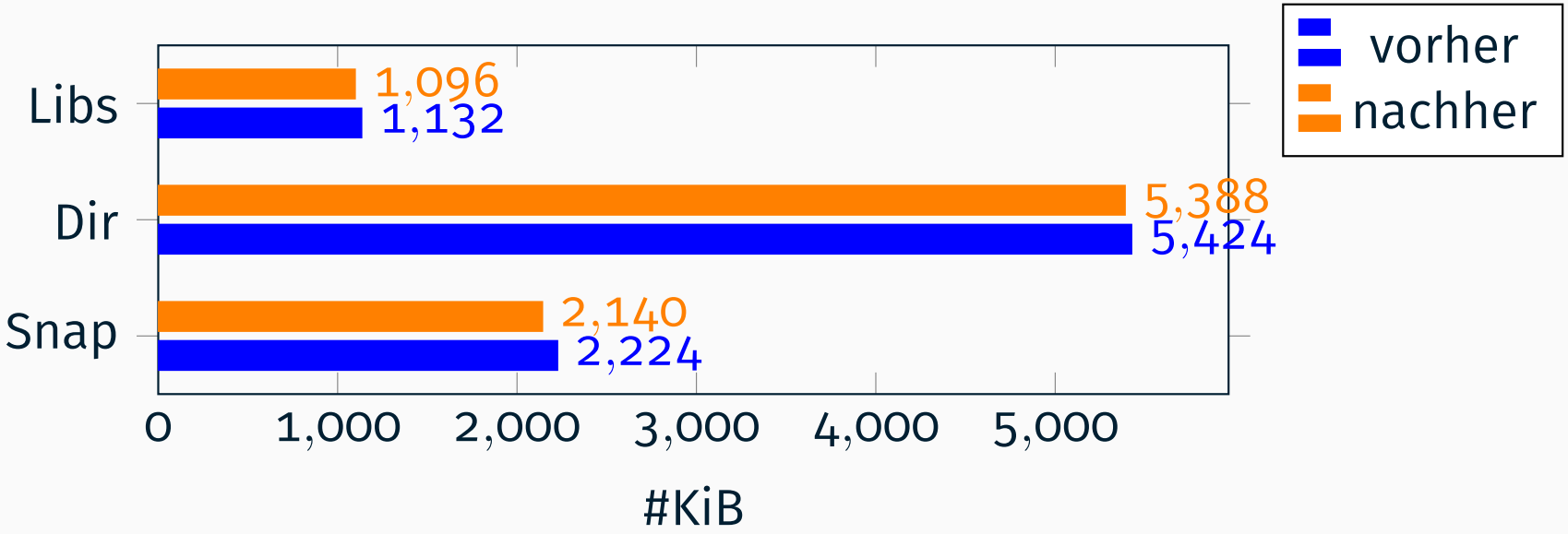


evince

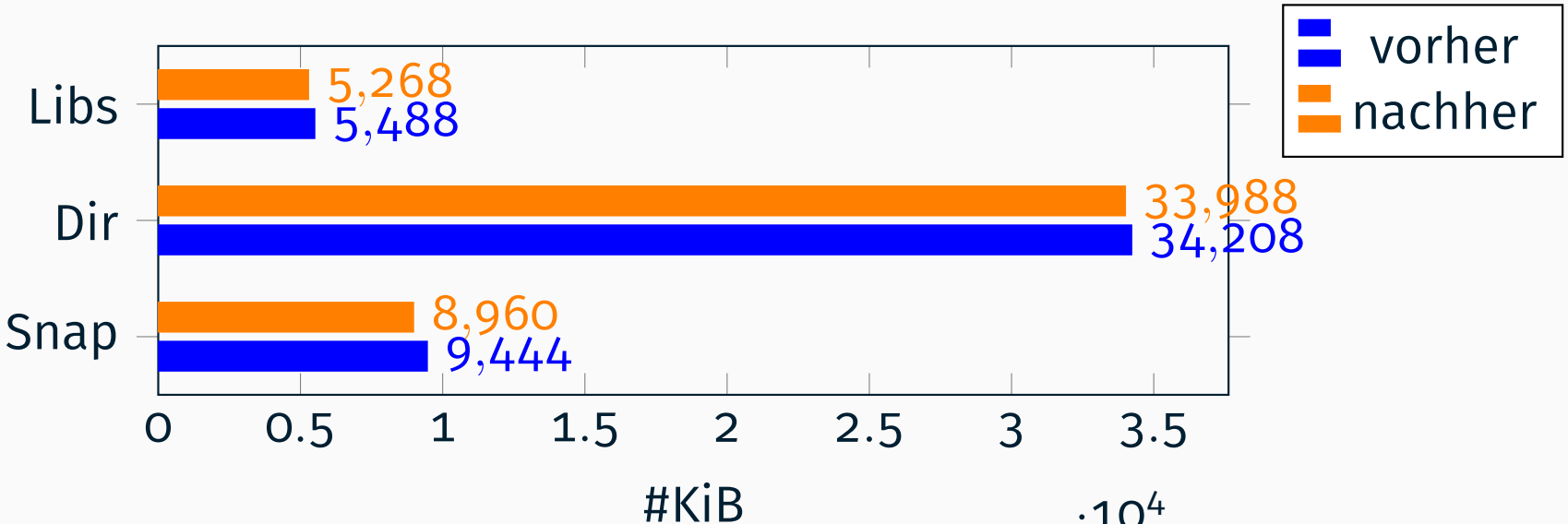


Statistiken - links und nmap

links

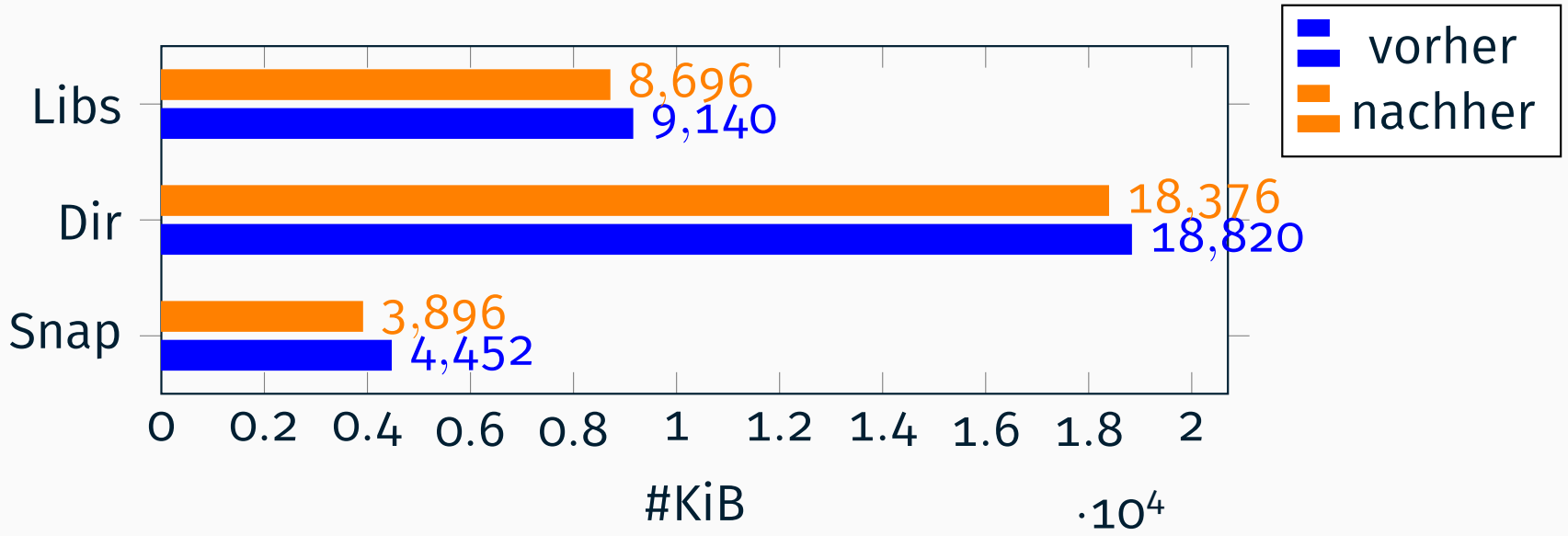


nmap

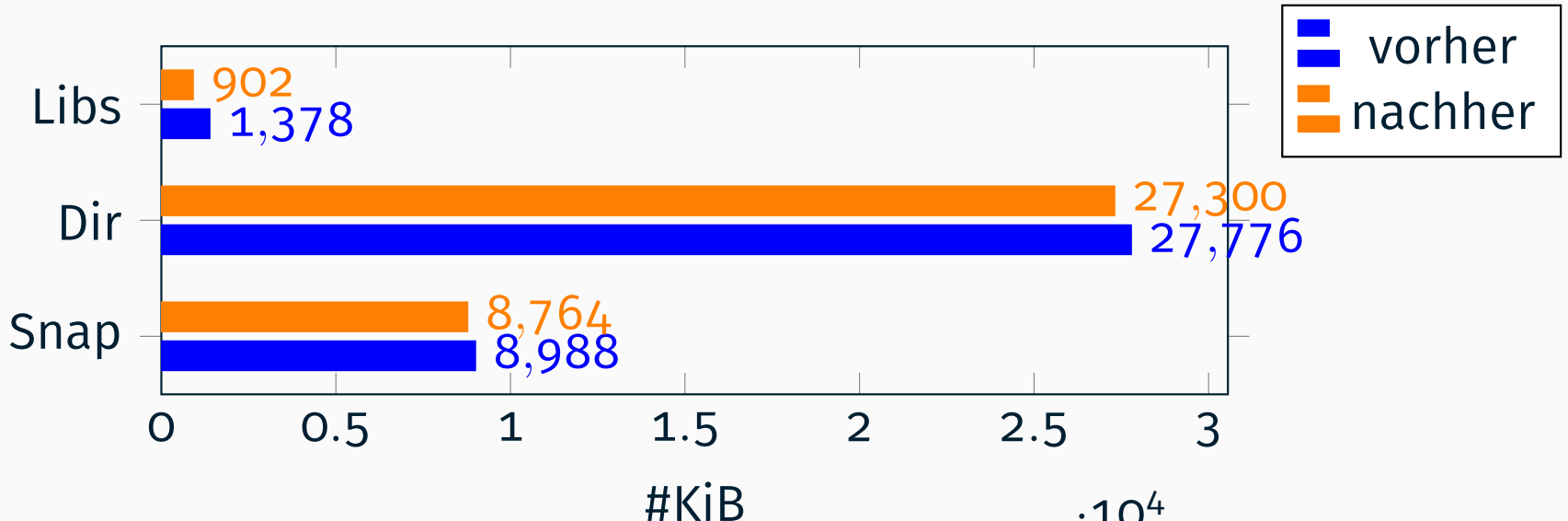


Statistiken - gnome-system-monitor und gnome-clocks

monitor

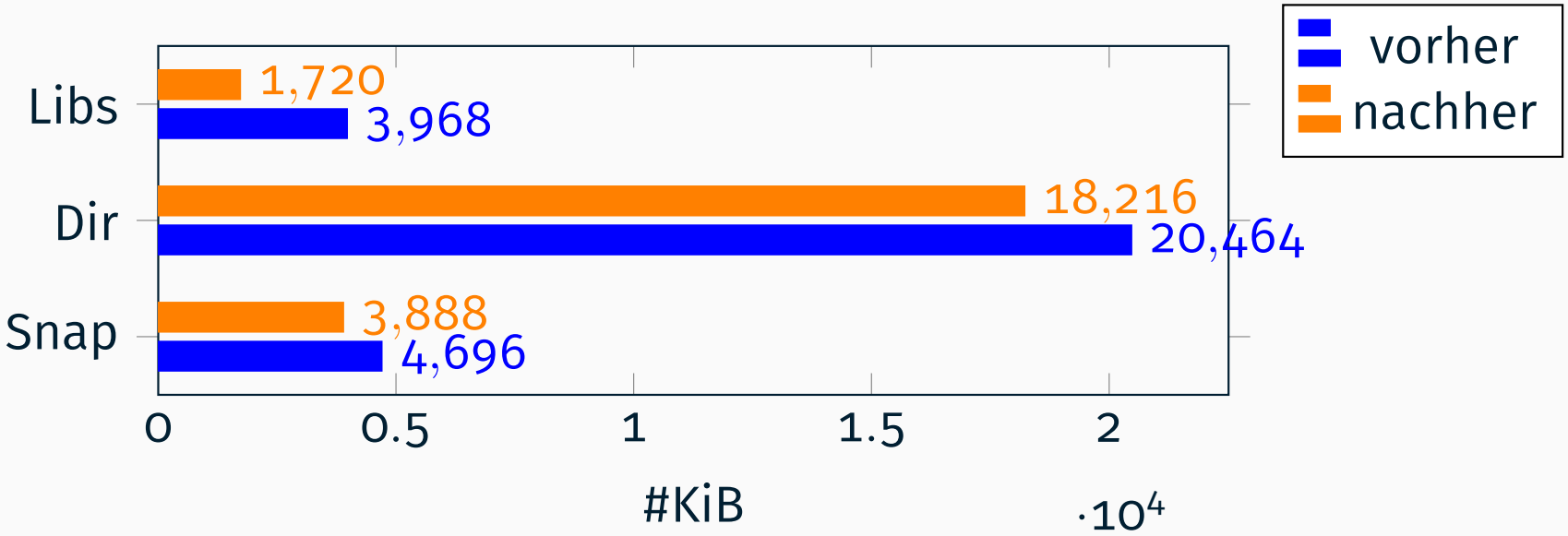


clocks

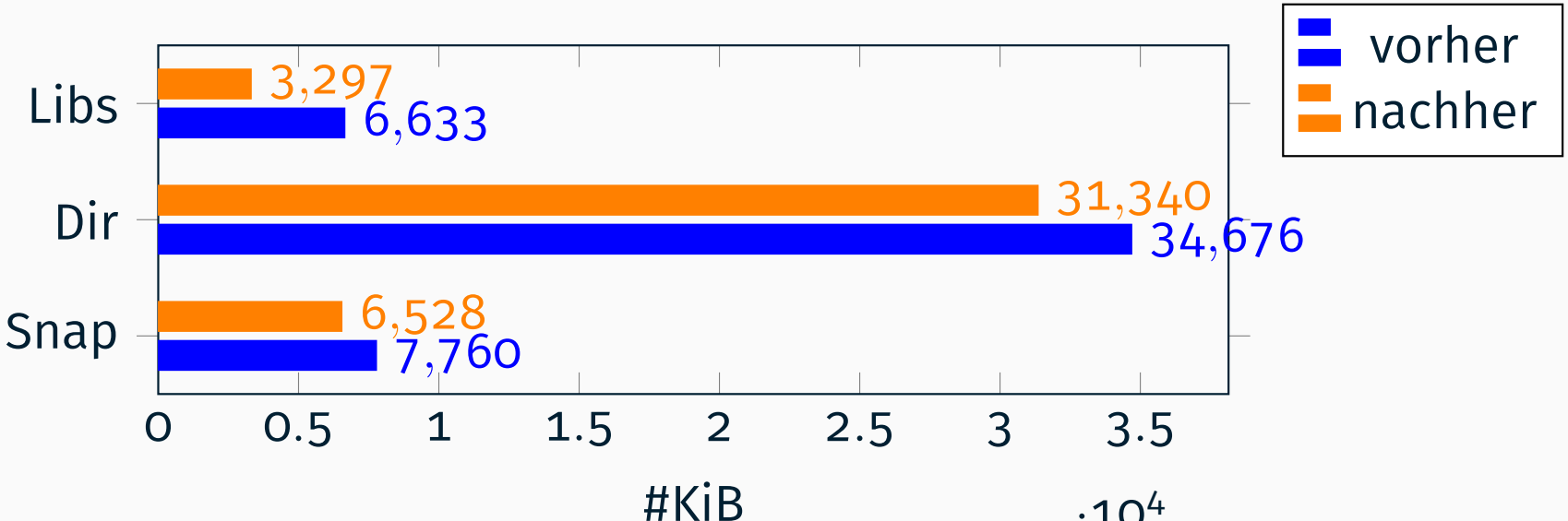


Statistiken - gnome-calculator und gedit

calculator



gedit



Ziele erreicht?

„Honey, I shrunk the ELFs”:

1. Statische Analyse
2. Dynamische Analyse
3. Überschreiben nicht genutzter Symbole
4. Löschen nicht genutzter Seiten

Ziele erreicht?

„Honey, I shrunk the Snaps”:

1. Umgebungsvariablen setzen, Debug-Symbole suchen
2. Zusätzliche Entrypoints?
3. Statische Analyse
4. Uprobes anpassen
5. Dynamische Analyse
6. Überschreiben nicht genutzter Symbole
7. Löschen nicht genutzter Seiten
8. Snap schrumpfen und neu verpacken

Ziele erreicht?

„Honey, I shrunk the Snaps”:

1. Umgebungsvariablen setzen, Debug-Symbole suchen
2. Zusätzliche Entrypoints?
3. Statische Analyse
4. Uprobes anpassen
5. Dynamische Analyse
6. Überschreiben nicht genutzter Symbole
7. Löschen nicht genutzter Seiten
8. Snap schrumpfen und neu verpacken

Ziele des Projekts:

- Anwenden der Toolchain auf Snap-Pakete
- Auswertung: Wie sehr lassen sich Snap-Pakete schrumpfen?
- Reduzierte Snap-Pakete neu verpacken

Ziele erreicht?

„Honey, I shrunk the Snaps”:

1. Umgebungsvariablen setzen, Debug-Symbole suchen
2. Zusätzliche Entrypoints?
3. Statische Analyse
4. Uprobes anpassen
5. Dynamische Analyse
6. Überschreiben nicht genutzter Symbole
7. Löschen nicht genutzter Seiten
8. Snap schrumpfen und neu verpacken

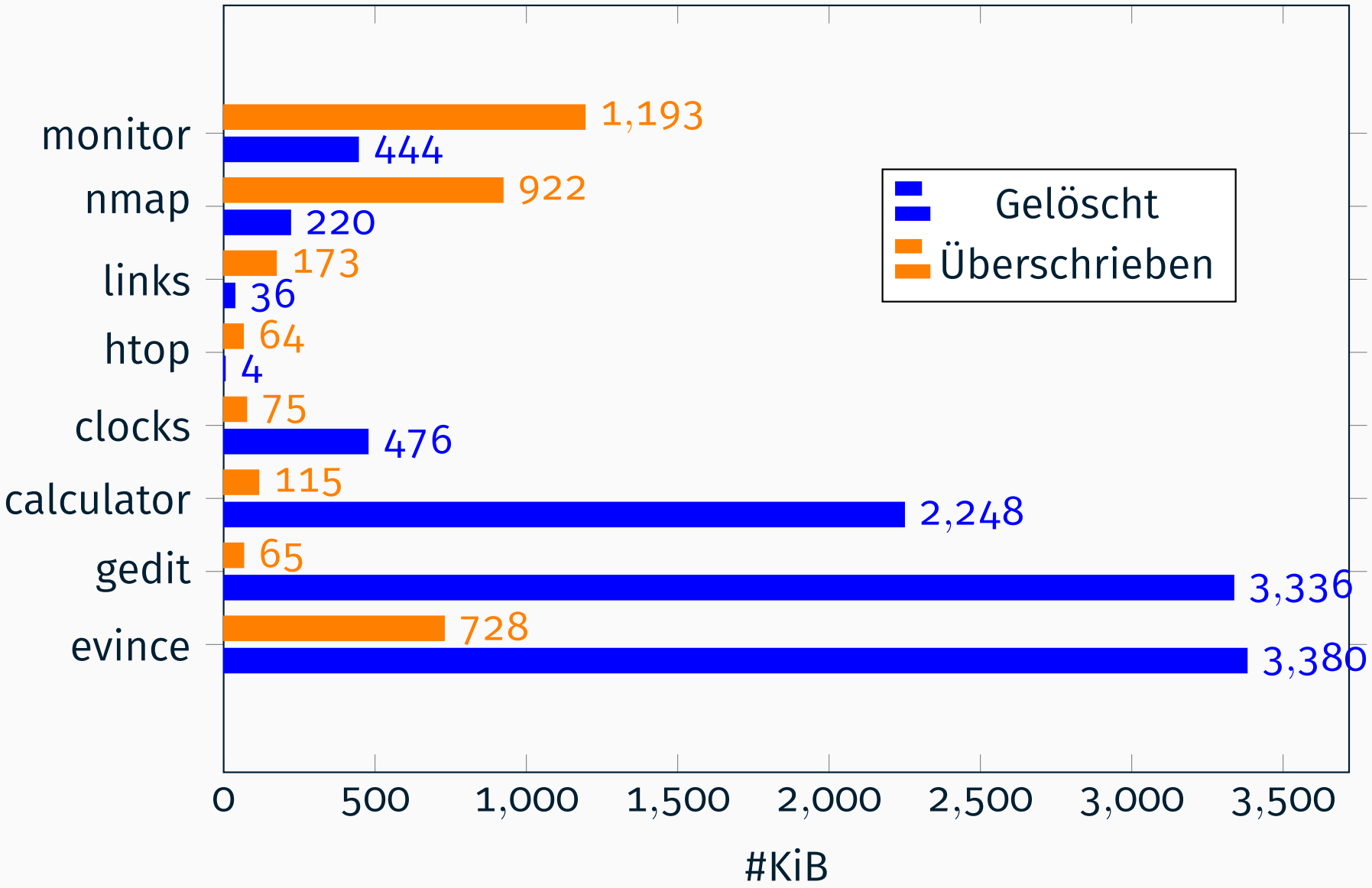
Ziele des Projekts:

- ✓ Anwenden der Toolchain auf Snap-Pakete
- ✓ Auswertung: Wie sehr lassen sich Snap-Pakete schrumpfen?
- ✓ Reduzierte Snap-Pakete neu verpacken

Fazit

- Unterschiede zwischen Snap-Paketen erschweren automatische Analyse
  - verschiedener Aufbau der Ordner-Struktur
  - unterschiedliche Aufrufketten
  - nicht nur im Binärformat ausgeliefert (Python, Node.js, …)
- Uprobes sind wichtig, aber
  - brauchen lange beim Aktivieren
  - haben sich zwischen Linux-Kernel-Versionen geändert
  - verlangsamen das System merklich
  - brauchen lange beim Deaktivieren

Gelöscht vs Überschrieben



Ausblick

- Bibliotheken lassen sich noch weiter schrumpfen
  - passende Pages zusammenfassen
  - derzeit laufende Masterarbeit
- Snaps bieten „Plugs“
  - doch wieder Abhängigkeiten zwischen Snap-Paketen
  - können bei Analyse mit berücksichtigt werden
  - werden erst zur Laufzeit in Namespace des Prozesses eingebunden

Danke für die Aufmerksamkeit!

Fragen?