

2 Threads & UNIX-Semantik F.9 UNIX — Prozesse, LWP's & Threads

★ **fork-Semantik**

- Problem
 - ◆ alle Threads duplizieren?
 - ◆ nur den aufrufenden Thread duplizieren?
- Lösung im Pthreads-Standard
 - ◆ im neu erzeugten Prozeß existiert nur der Thread, der fork aufgerufen hat
- Alternative bei Solaris-Threads
 - ◆ **Fork One Model:**
nur der aufrufende Thread wird in den Sohnprozeß übernommen
 - ◆ **Fork All Model:**
Sohnprozeß enthält alle Threads

AKBP |

Ausgewählte Kapitel der praktischen Betriebsprogrammierung
© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

F-Prozesse.doc 1999-01-20 09.04

F.53

Reproduktion jeder Art oder Vervielfältigung dieser Unterlagen, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

2 Threads & UNIX-Semantik (3) F.9 UNIX — Prozesse, LWP's & Threads

★ **Signal-Semantik (2)**

- ▲ Signal-Zustellung an Threads
- Lösung unterscheidet nach der Art der Signal-Entstehung
 - ◆ Traps: synchron durch die Programmausführung erzeugt (Segmentation fault, Illegal instruction, ...)
 - Signal wird an den verursachenden Thread zugestellt
 - ◆ Explizite Signalerzeugung durch das Programm (Funktion *pthread_kill*)
 - Signal wird an den angegebenen Ziel-Thread zugestellt
 - ◆ Interrupts: asynchron von "außen" erzeugt (Interrupt, Quit, Hangup, SIGIO, ...)
 - Signal wird dem gesamten Prozess zugestellt
-

AKBP |

Ausgewählte Kapitel der praktischen Betriebsprogrammierung
© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

F-Prozesse.doc 1999-01-20 09.04

F.55

Reproduktion jeder Art oder Vervielfältigung dieser Unterlagen, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

2 Threads & UNIX-Semantik (2) F.9 UNIX — Prozesse, LWP's & Threads

★ **Signal-Semantik**

- Grundlegendes Signal-Behandlungskonzept unverändert:
 - Signal ignorieren / Default-Reaktion / Signal abfangen
 - ➔ Einstellung gilt immer für alle Threads eines Prozesses
- Problem
 - ◆ welchem Thread wird ein eintreffendes Signal zur Bearbeitung zugestellt

AKBP |

Ausgewählte Kapitel der praktischen Betriebsprogrammierung
© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

F-Prozesse.doc 1999-01-20 09.04

F.54

Reproduktion jeder Art oder Vervielfältigung dieser Unterlagen, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.

2 Threads & UNIX-Semantik (3) F.9 UNIX — Prozesse, LWP's & Threads

★ **Signal-Semantik (3)**

- Signal-Masken
 - ◆ Signale können in UNIX maskiert werden — die Zustellung zurückgehalten
- Signal-Masken werden von Threads an neu erzeugte Threads vererbt
- Threads können Thread-spezifische Signal-Masken setzen (Funktion *pthread_sigmask*)
- ▲ Bearbeitung von Signalen an den gesamten Prozess
 - ◆ Zustellung erfolgt an einen der Threads, die das Signal nicht blockiert haben
 - ◆ Auswahl des Threads erfolgt zufällig

AKBP |

Ausgewählte Kapitel der praktischen Betriebsprogrammierung
© Jürgen Kleinöder, Universität Erlangen-Nürnberg, IMMD IV, 1999

F-Prozesse.doc 1999-01-20 09.04

F.56

Reproduktion jeder Art oder Vervielfältigung dieser Unterlagen, außer zu Lehrzwecken an der Universität Erlangen-Nürnberg, bedarf der Zustimmung des Autors.