

Signale

PD Dr. Reinhard Bündgen
buendgen@de.ibm.com

Signale in Unix (POSIX)

- Kommunikation mit Prozessen
 - Prozess zu Prozess
 - Kern zu Prozess
 - wenig Information: kl. Zahl: 1-64 (etwas Verursacherinfo)
- Signale
 - siehe `<asm/signal.h>`
 - man `signal.h`, `/usr/include/signal.h`
 - z.B. HUP, INT, KILL, SEGV
 - reguläre Signale, RT Signale
- Signalbehandlung
 - asynchron
 - durch Kern (ignore, stop, continue, terminate, dump)
 - durch Prozess: Signalbehandlungsroutine

SIGNAL User Space Interface

- Kommando: `kill` sendet Signal
- API
 - `kill`, `tkill()`, `tgkill()`
 - sende Signal an Prozess (thread group), Thread
 - `sigaction()`, `signal()`
 - installiere Signalbehandlungsroutine
 - `sigprocmask()`
 - ändere Menge der zu blockierenden Signale
 - `sigpending()`, `sigsuspend()`
 - teste warte auf Signal
 - `rt_sigaction`, `rt_sigpending`, `rt_sigprocmask`, ...
 - RT Varianten

Signaleigenschaften

- Signalbehandlung
 - Defaulthandler – vom Kern festgelegt
 - vom User Space abfangbar/überschreibbar durch User Space Handler oder SIG_IGN
- blockierbar: temporäre Maskierung
- maskierbar während Signalbehandlung
- Signalqueueing nur für RT Signale (>31) möglich
- benötigt Authorisierung (capabilities)
- Anzahl ausstehender Signale/Prozess begrenzt
- POSIX 1003.1 Anforderungen an multithreaded Prozesse:
 - Signal Handler gelten prozessweit (shared)
 - Kern und kill() senden Signale an einen Prozess, Behandlung durch beliebigen das Signal nicht blockierenden Thread
 - fatale Signale terminieren alle Threads eines Prozesses

Kernel Aufgaben

- Signal-Erzeugung
 - markiere eine Task, dass sie ein Signal erhalten hat
 - speichere signal pending info in `task_struct`
 - teste ob Signal blockiert ist, wenn nein
 - setze `TIF_SIGPENDING` flag in `thread_info->flags`
 - evtl. wecke task aus `TASK_INTERRUPTIBLE` oder `TASK_STOPPED` Status auf.
- Signal-Auslieferung
 - lasse Task das Signal behandeln
 - durch ignorieren
 - durch default Handler
 - durch User Space Handler

Kernel Datenstrukturen

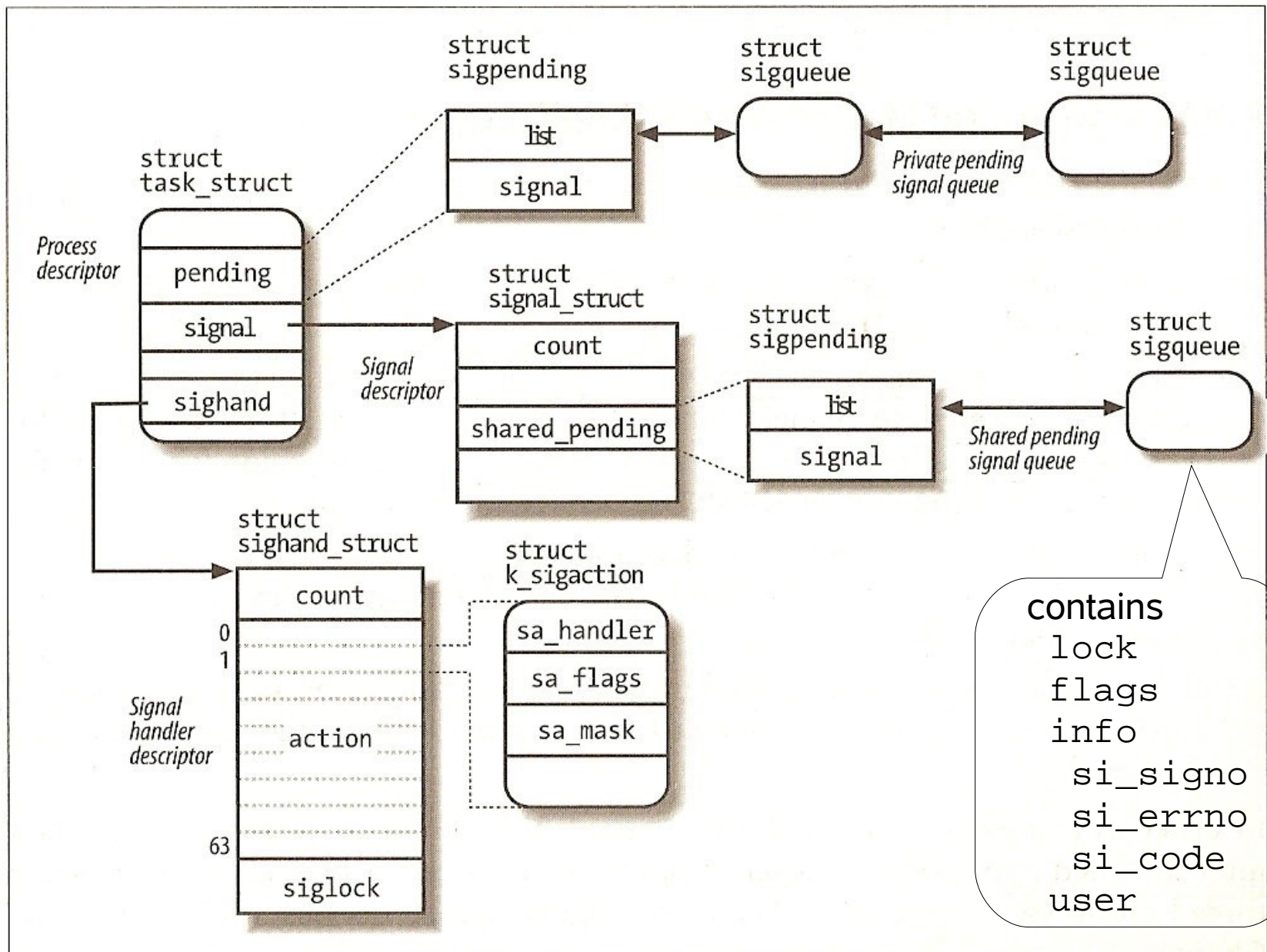


Figure 11-1. The most significant data structures related to signal handling

User Space Signal Handler

