Übungen zu Vorlesung
Safety-Critical
Systems 1
Sommersemester 2002

AG BS
Jan Peleska

Serie 2

Aufgabe 1: Partielle Verifikation mit Fehlerbäumen

Wenden Sie die Methode der Fehlerbaumanalyse an, um eine partielle Softwareverifikation des Linux Schedulers durchzuführen [Dateien sched.c, der eigentliche Scheduler ist dort in der C-Function schedule() definiert, Date sched.h enthält Typ- und Makrodefinitionen]. Das Ziel der partiellen Verifikation ist, die Software hinsichtlich folgender "Bedrohung" zu untersuchen:

> Potentieller Software Hazard im Linux Scheduler:
> Ein Prozess vom Scheduling-Typ SCHED_FIFO hat höchste Priorität und ist rechenbereit. Der Scheduler gibt die CPU jedoch einem anderen Prozess vom Scheduling Typ SCHED_OTHER.

1.1 Untersuchen Sie den Programmcode und identifizieren Sie mit Hilfe der Fehlerbaumanalyse, welche Softwareanteile in sched.c bzw Makros in sched.h zu diesem Hazard beitragen könnten. Dabei können Sie davon ausgehen, dass folgende in schedule() aufgerufene Funktionen *keinen* Beitrag zum Entstehen des Hazards liefern: run_task_queue(), do_bottom_half(), get_mmu_context(), switch_to(). Weiterhin dürfen Sie annehmen, dass die verschiedenen Lock-Funktionen korrekt implementiert sind.

1.2 Führen Sie die Fehlerbaumanalyse für die kritischen Softwareanteile bis auf die Ebene einzelner Kontrollstrukturen (if, while, for, switch) und Zuweisungen durch, so dass man erkennen kann, ob der o.g. Hazard wirklich ausgeschlossen ist.

1.3 Erläutern Sie, welche weiteren Untersuchungen (ausserhalb des Scheduler Codes) noch zu machen wären, um die Abwesenheit des genannten Hazards vertrauenswürdig demonstrieren zu können.