Software Systems Engineering

Abschlussarbeit (BSc.)
Titel: Inkrementelle Dead-Code Analyse von BusyBox

Warum soll ich mich damit beschäftigen?

Bei einer Softwareproduktlinie spricht man von einer hochgradig konfigurierbaren Software um ohne Clone-and-Own Praktiken kunden-/markt-spezifische Versionen anbieten zu können. Ein bekanntes Beispiel stellt Windows 10 dar (welches unter Anderem in den folgenden Versionen angeboten wird: Home, Education, Pro, Enterprise, für die jeweils S, N oder SN Versionen optional verfügbar sind).
Die Uni Hildesheim hat für die Analyse von variablen Bestandteilen in Softwareproduktlinien das Werkzeug KernelHaven entwickelt. Dieses ist in der Lage verschiedene Produktlinienbezogene Analysen durchzuführen. Eine sehr zeitintensive Analyse stellt die sogenannte DeadCode Analyse dar, in der analysiert wird ob variabler Code tatsächlich so konfigurierbar ist, wie es von den Entwicklern angedacht ist. Eine Erweiterung für KernelHaven bietet eine drastische Beschleunigung dieser Analyse dar (Zeitbedarf sinkt auf etwa 10%). Diese Analyse wurde bereits erfolgreich auf dem Linux Kernel getestet.

Was soll ich tun?

In dieser Arbeit geht es konkret um die Inkrementelle Dead Code Analyse für BusyBox. Hierzu gibt es bereits eine initiale Implementierung um BusyBox mit KernelHaven analysieren zu können. Ihre Aufgabe ist es, KernelHaven so zu erweitern, dass die inkrementelle Analyse auf BusyBox angewendet werden kann.
Anschließend soll die Performanz der klassischen DeadCode Analyse und der inkrementellen DeadCode Analyse verglichen werden.

Welche Ergebnisse werden erwartet?

Folgende Ergebnisse werden mindestens erwartet:

- Konzeptuelle Anpassung von KernelHaven für die inkrementelle Analysen von Busybox
- Implementierung der inkrementellen Dead-Code-Analyse für Busy-Box
- Performanzanalyse der klassischen Dead-Code-Analyse und der inkrementellen Variante

Was bringt mir das?

Neben den Kreditpunkten und der Note für den oben genannten Typ dieser Arbeit:

- Einblicke in die Entwicklung und Evolution realer, großer Softwareprojekte
- Erfahrung mit rechenintensiven Anwendungen
- Vertiefung der Programmierkenntnisse in Java

Kontakt
M.Sc. Moritz Flöter
floeter@uni-hildesheim.de

Aufteilung der Arbeit

Theorie	*Implementierung*	*Literatur*
30%	60%	10%