

Gedächtnis-Protokoll Übersetzerbau 3 Prüfung

Datum: 04.03.2013

Prüfer: Stefan Kempf, Tobias Werth

Ergebnis: sehr gut

Vorbereitung

Ca. eineinhalb Wochen aus den Folien gelernt, selber Beispiele aus den Folien nach-“gerechnet“. Dabei habe ich mich auf die Hauptthemen JIT, Garbage Collection, Linker&Loader und Exceptions konzentriert. Die Woche vorher war das Compilerbau-Praktikum.

Themen

Die Prüfung war stark auf das vorherige Praktikum ausgerichtet - JIT war der Hauptteil der Prüfung. Dann kamen noch Fragen zu dyn. Bibliotheken und Exceptions. Es können natürlich auch Fragen zu den anderen Themen kommen. Aspekte sind sicher ein heisser Kandidat.

Prüfungsverlauf

Fragen wurden von Stefan gestellt.

Einstieg: *Was gibt es für Methoden Bytecode zu interpretieren?*

Einfaches Interpretieren, indirektes und direktes Durchfädeln. Erklärt warum das langsam ist (Sprünge, Cache, calls, ...)

Wie macht man es dann richtig?

Natives Übersetzen. Probleme mit Sprungzielen (CISC Code, indirekte Sprünge). Warum bei JIT-Sprachen eben Befehle fester Größe hat, nur relative Sprünge. Wegfall der Portabilität. Nutzung des CPU-I-Cache.

Wie macht der JIT das mit Funktionsaufrufen?

→ Die Frage zielte auf Erkennung von Rekursion ab. Wird durch Trampolin gelöst (wird im Praktikum gemacht).

Was gibt es noch für Probleme beim JIT?

Code-Cache Verwaltung, warum LRU nicht toll ist, Sprünge zwischen Blöcken, Fragmentierung, ...

Wie geht Linear Scan?

Grundstruktur mit zwei Läufen, einer zum bauen Lebendigkeitsintervalle und einer zur Registervergabe (wird auch nochmal im Praktikum gemacht bzw. implementiert).

Beispiel vorrechnen

Fragen zur Heuristik: *Warum verdrängt man man das Intervall mit dem am weitest hinten liegenden Ende?*

Um... *auf-dem-Schlauch-steh*

Was macht man denn beim Graphfärben?

Man verdrängt die Variable in den Speicher die mit den meisten anderen kollidiert. Erklärt warum das nicht immer optimal ist (siehe andere Heuristiken in den Folien).

Wie werden Funktionsaufrufe bei dyn. Bibliotheken gemacht?

PLT, lazy-loading, Überschreiben des PLT-Eintrags.

Könnte man das für den JIT aus dem Praktikum auch tun?

Ja, PLT-Tabelle mit default eintrag mit Zeiger auf Trampolin und dann überschreiben mit Zeiger auf native Code.

Was muss man tun wenn man Native-Code und Interpretieren mischen will?

Native->Interpretiert: Interpreter-Kontext aufbauen mit Parametern, evtl. müssen Typen konvertiert werden (z.b. int32<->int64), Referenzen auf arrays müssen behandelt werden, rückgabewert zurückwandeln, etc ...

Was für Verfahren für das Behandeln von Exceptions kamen in der Vorlesung dran?

Stack-cutting (erklärt wie setjmp, longjmp funktionieren). Statische Tabellen (Intervall-Tabelle, Exception-Typen, Handler).

Aber was ist mit Funktionsaufrufen?

Stack-unwinding: Bei throw muss erst in der Tabelle von der Funktion die oben auf dem Stapel liegt nachgeschaut werden, dann in der darüber, ... bis ein handler gefunden wird oder eben abort.

Fazit

Die Atmosphäre war sehr angenehm. Man muss die Themen schon verstanden und durchdacht haben, damit man auch Beispiele und Transferfragen beantworten kann. Keine „gemeinen“ Fragen. Insgesamt absolut fair.