

Masterarbeit

Titel der Masterarbeit

Phil Steinhorst

<i>Erstgutachter und Betreuung</i>	Prof. Dr. Jan Vahrenhold
<i>Zweitgutachter</i>	Prof. Dr. Markus Müller-Olm

Münster, 14. November 2018

Titel der Masterarbeit

Masterarbeit zur Erlangung des akademischen Grades *Master of Education*
in den Fächern Mathematik und Informatik

Erstgutachter und Betreuung: Prof. Dr. Jan Vahrenhold

Zweitgutachter: Prof. Dr. Markus Müller-Olm

Münster, 14. November 2018

Phil Steinhorst

Dürerstraße 1, 48147 Münster

p.st@wwu.de

Matrikelnummer: 382 837

Westfälische Wilhelms-Universität Münster

Fachbereich 10 – Mathematik und Informatik

Institut für Informatik

Einsteinstraße 62, 48149 Münster

Zusammenfassung

Die vorliegende Arbeit soll eine Aufarbeitung verschiedener Ansätze für Garbage-Collection-Algorithmen liefern. Nach einer kurzen Darstellung der zugrunde liegenden Problematik und deren praktische Relevanz sowie den Vor- und Nachteilen einer automatischen Speicherverwaltung gegenüber einer manuellen Speicherverwaltung werden gängige Ansätze vergleichend vorgestellt sowie Einsatz und Eignung in der Praxis beurteilt. Als Gütekriterien dienen hier beispielsweise Laufzeitbetrachtungen, Speicherbedarf und entstehende Verzögerungen im Programmablauf, die für ausgewählte Ansätze besonders detailliert untersucht werden.

Weiter wird eine Anwendung entworfen, mit der die Arbeitsweise der diskutierten Garbage-Collection-Ansätze visualisiert werden kann. Dazu gehört eine angemessene Visualisierung eines beschränkten Speicherbereichs, etwa durch eine optische Unterscheidbarkeit belegter Blöcke, sowie der einzelnen Arbeitsphasen, die eine Garbage Collection ausführt. Dabei sollen auch unterschiedliche Szenarien auswählbar sein, etwa verschiedene Speicherfüllstände und eine variable Anzahl bzw. Größe von Objekten, die im Speicher hinterlegt sind.

Am Ende nochmal schauen, ob das wirklich so ist :D

Abstract

Englisch einfügen.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Problemstellung und Terminologie	1
I	Algorithmen und Ansätze	3
2	Mark and Sweep	5
2.1	Naives Mark and Sweep	5
II	Entwurf und Realisierung eines Garbage-Collection-Simulators	7
3	Modellierung	9
	Anhang	11
A	Test	13
	Literatur	15
	Abbildungsverzeichnis	17
	Tabellenverzeichnis	19
	Eigenständigkeitserklärung	21

Einleitung

Einleitung: Was ist Garbage Collection, wozu ist das relevant?

1.1 Problemstellung und Terminologie

Was ist das Ziel einer GC? Wie kann man das möglichst formal ausdrücken? Grundbegriffe und Modellierung des Speichers?

Teil I

Algorithmen und Ansätze

Mark and Sweep

Wir beginnen mit einer Vorstellung des ersten Garbage-Collection-Algorithmus, der auf John MCCARTHY zurückgeht [McC60, S. 191–193]. Im Rahmen eines im Jahr 1960 veröffentlichten Artikels über die Berechnung rekursiver Funktionen auf dem *IBM 704* mithilfe des *LISP Programming Systems* erläutert McCarthy die Speicherung von Daten in einer Listenstruktur. Diese besteht aus Paaren, deren erster Eintrag *car* die zu speichernde Information enthält, während im zweiten Eintrag *cdr* die Registeradresse des nachfolgenden Paares zu finden ist. Register, die aktuell nicht zur Speicherung von Daten genutzt werden, befinden sich in einer *free storage list*. Bei der Anforderung von Speicher für ein zu speicherndes Datum werden Register aus dieser Liste entfernt. Durch die Manipulation der Registeradressen können Paare verwaisen, was zu Speicherlecks führt. Zur Auflösung dieser Problematik bietet LISP als erste Programmiersprache ihrer Zeit eine automatische Speicherverwaltung, die von McCarthy wie folgt grob umschrieben wird: Im Falle von Speicherknappheit wird – ausgehend von einer Menge von Basisregistern – ermittelt, welche Register über eine Folge von *cdr*-Einträgen erreichbar sind. Nicht erreichbare Register enthalten überschreibbare Inhalte, sodass diese zurück in die *free storage list* eingefügt werden können und wieder als freie Speicherplätze zu Verfügung stehen. Diese zweischrittige Vorgehensweise – das Erkennen nicht mehr benötigter Speicherbereiche und die anschließende Freigabe eben jener – bildet die Grundlage des *Mark-and-Sweep*-Algorithmus.

2.1 Naives Mark and Sweep

Teil II

Entwurf und Realisierung eines
Garbage-Collection-Simulators

Modellierung

3

Designentscheidungen

Anhang

Test

A

blablubb

Literatur

- [McC60] John McCarthy. „Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I“. In: *Communications of the ACM* 3.4 (1960), S. 184–195 (zitiert auf Seite 5).

Diese Masterarbeit wurde mit L^AT_EX 2_ε unter Verwendung der Vorlage *Clean Thesis* von Ricardo Langner gesetzt. Für mehr Informationen siehe <http://cleanthesis.der-ric.de/>.

Abbildungsverzeichnis

Tabellenverzeichnis

Eigenständigkeitserklärung

Hiermit versichere ich, dass die vorliegende Masterarbeit *Titel der Masterarbeit* selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

(Ort, Datum)

(Unterschrift)