

**Akademia Górniczo-Hutnicza
im. Stanisława Staszica w Krakowie**

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

KATEDRA INFORMATYKI STOSOWANEJ



PRACA INŻYNIERSKA

KAJETAN RZEPECKI

**Implementacja maszyny wirtualnej dla funkcyjnych
języków programowania wspierających przetwarzanie
współbieżne.**

PROMOTOR:

dr inż. Piotr Matyasik

Kraków 2013-10-19

OŚWIADCZENIE AUTORA PRACY

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMNIENIONE W PRACY.

.....

PODPIS

AGH
University of Science and Technology in Krakow

Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering
DEPARTMENT OF APPLIED COMPUTER SCIENCE



BACHELOR OF ENGINEERING THESIS

KAJETAN RZEPECKI

**Implementation of a virtual machine for functional
programming languages with support for concurrent
computing.**

SUPERVISOR:
Piotr Matyasik, Ph.D

Krakow 2013-10-19

Spis treści

1	Motywacja pracy	6
2	Implementacja ThesisVM	7
2.1	Reprezentacja pośrednia programów	7
2.2	Reprezentacja obiektów ThesisVM	7
2.3	Reprezentacja stanu ThesisVM	7
2.4	Funkcja zmiany stanu ThesisVM	7
3	Zarządzanie Pamięcią	8
3.1	Organizacja pamięci ThesisVM	8
3.2	Alokacja obiektów	8
3.3	Kolekcja nieosiągalnych obiektów	8
3.4	Kolekcja obiektów cyklicznych	8
4	Model Przetwarzania Współbieżnego	9
4.1	Model Aktorowy	9
4.2	Notacja procesu w ThesisVM	9
4.3	Harmonogramowanie procesów	9
4.4	Przesyłanie wiadomości	9
A	Kompilator kodu bajtowego	10
B	Wizualizacja stanu maszyny wirtualnej	11
C	Przykładowe przebiegi programów	12
D	Analiza wydajności ThesisVM	13

1. Motywacja pracy

Celem pracy i powiązanego z nią projektu jest implementacja oraz ewaluacja maszyny wirtualnej dla funkcyjnych języków programowania, które umożliwiają przetwarzanie współbieżne wykorzystując Model Aktorowy ([?]) oraz asynchroniczne przekazywanie wiadomości.

- Opisać problemy Erlanga,
- opisać próby ich rozwiązania w Erlangu,
- opisać proponowany sposób ich rozwiązania,
- umotywować powstanie ThesisVM.

2. Implementacja ThesisVM

2.1. Reprezentacja pośrednia programów

2.2. Reprezentacja obiektów ThesisVM

2.3. Reprezentacja stanu ThesisVM

2.4. Funkcja zmiany stanu ThesisVM

3. Zarządzanie Pamięcią

3.1. Organizacja pamięci ThesisVM

3.2. Alokacja obiektów

3.3. Kolekcja nieosiągalnych obiektów

3.4. Kolekcja obiektów cyklicznych

4. Model Przetwarzania Współbieżnego

4.1. Model Aktorowy

4.2. Notacja procesu w ThesisVM

4.3. Harmonogramowanie procesów

4.4. Przesyłanie wiadomości

A. Kompilator kodu bajtowego

Opisać pipeline kompilatora.

B. Wizualizacja stanu maszyny wirtualnej

Opisać narzędzie do rysowania grafów stanu.

C. Przykładowe przebiegi programów

Dać kilka przykładów prostych programów razem z grafami stanów.

D. Analiza wydajności ThesisVM

Przeanalizować wydajność GC i SMP.

Bibliografia

[Tes00] T. Test. *Test title of a test BibTeX position*. Test, Test, 2000.