

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

Wydział Elektrotechniki, Automatyki, Informatyki i Inżynierii Biomedycznej

KATEDRA INFORMATYKI STOSOWANEJ



AGH

PRACA INŻYNIERSKA

KAJETAN RZEPECKI

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

PROMOTOR:

dr inż. Piotr Matyasik

Kraków 2013

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 WYMIENIONE 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

Spis treści

1	Wstęp	8
1.1	Problemy przetwarzania współbieżnego	8
1.2	Próby rozwiązania problemu	8
1.3	Cel i zawartość pracy	8
2	Projekt i implementacja ThesisVM	9
2.1	Reprezentacja pośrednia programów	9
2.2	Reprezentacja prostych obiektów ThesisVM	9
2.3	Reprezentacja obiektów funkcyjnych ThesisVM	9
2.4	Reprezentacja kodu bajtowego ThesisVM	9
2.5	Ewaluacja argumentów i aplikacja funkcji	9
2.6	Operacje arytmetyczne	9
2.7	Implementacja wbudowanych operatorów	9
2.8	Kompilator kodu bajtowego ThesisVM	9
3	Model zarządzanie pamięcią	10
3.1	Organizacja pamięci ThesisVM	10
3.2	Alokacja obiektów	10
3.3	Kolekcja nieosiągalnych obiektów	10
3.4	Kolekcja obiektów cyklicznych	10
4	Model przetwarzania współbieżnego	11
4.1	Model Aktorowy	11
4.2	Notacja procesu w ThesisVM	11
4.3	Harmonogramowanie procesów	11
4.4	Przesyłanie wiadomości	11
5	Podsumowanie i analiza wydajności ThesisVM	12
5.1	Leniwe zliczanie referencji	12
5.2	Przesyłanie wiadomości	12

5.3 Porównanie szybkości działania ThesisVM	12
A Wizualizacja stanu maszyny wirtualnej	13
B Przykładowe programy	14

Spis rysunków

Spis tablic

1. Wstęp

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 ¹.

1.1. Problemy przetwarzania współbieżnego

- Opisać problemy Erlanga,

1.2. Próby rozwiązania problemu

- opisać próby ich rozwiązania w Erlangu,

1.3. Cel i zawartość pracy

- opisać proponowany sposób ich rozwiązania,
- umotywować powstanie ThesisVM.

¹Test

2. Projekt i implementacja ThesisVM

TODO: Opisać ogólną strukturę maszyny wirtualnej (z GC i SMP) i opisać o czym będzie niniejsza sekcja.

2.1. Reprezentacja pośrednia programów

2.2. Reprezentacja prostych obiektów ThesisVM

2.3. Reprezentacja obiektów funkcyjnych ThesisVM

2.4. Reprezentacja kodu bajtowego ThesisVM

2.5. Ewaluacja argumentów i aplikacja funkcji

2.6. Operacje arytmetyczne

2.7. Implementacja wbudowanych operatorów

2.8. Kompilator kodu bajtowego ThesisVM

Opisać pipeline kompilatora.

3. Model 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

5. Podsumowanie i analiza wydajności ThesisVM

Przeanalizować wydajność GC i SMP.

5.1. Leniwe zliczanie referencji

Przeanalizować szybkość, pauzy, zużycie pamięci.

5.2. Przesyłanie wiadomości

Przeanalizować szybkość przesyłania wiadomości/konieczność czekania procesów/wątków.

5.3. Porównanie szybkości działania ThesisVM

Porównać kilka implementacji prostych programów (z Haskell'em, leniwym Lispem itp).

A. Wizualizacja stanu maszyny wirtualnej

Opisać narzędzie do rysowania grafów stanu.

B. Przykładowe programy

Dać kilka przykładów prostych programów razem z grafami stanów.

Bibliografia

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