



Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie

**WYDZIAŁ ELEKTROTECHNIKI, AUTOMATYKI,
INFORMATYKI I INŻYNIERII BIOMEDYCZNEJ**

KATEDRA INFORMATYKI STOSOWANEJ

Praca dyplomowa inżynierska

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

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

Autor:	<i>Kajetan Rzepecki</i>
Kierunek studiów:	<i>Informatyka</i>
Opiekun pracy:	<i>dr inż. Piotr Matyasik</i>

Kraków, 2014

*Oświadczam, świadomy odpowiedzialności karnej za poświadczenie nie-
prawdy, że niniejszą pracę dyplomową wykonałem osobiście i samodzielnie
i nie korzystałem ze źródeł innych niż wymienione w pracy.*

Serdecznie dziękuję Lucynie za cierpliwość i wsparcie podczas tworzenia pracy.

Spis treści

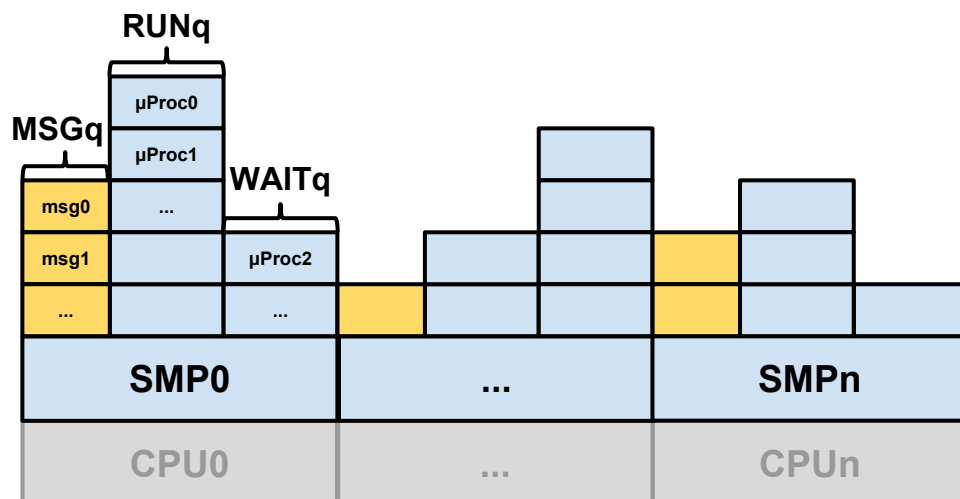
1. Model przetwarzania współbieżnego	7
1.1. Implementacja symetrycznego multiprocessingu	7
1.2. Harmonogramowanie procesów	8
1.3. Implementacja Modelu Aktorowego	8
1.4. Implementacja przesyłania wiadomości	9

1. Model przetwarzania współbieżnego

- Opisać bardziej szczegółowo model SMP.
- Opisać bardziej szczegółowo Model Aktorowy i asynchroniczne przekazywanie wiadomości. [?, ?]

1.1. Implementacja symetrycznego multiprocessingu

- Opisać bardziej szczegółowo implementację SMP

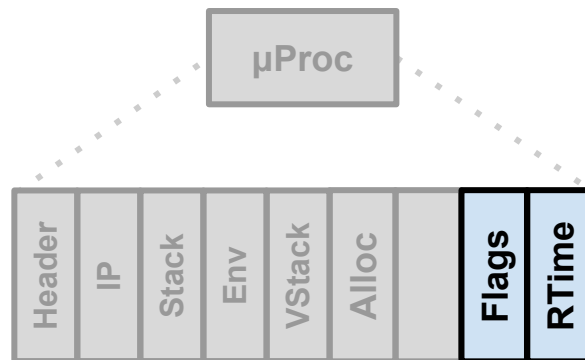


Rysunek 1.1: Schemat symetrycznego multiprocesora ThesisVM.

- Co SMP robi podczas interpretacji kodu.
- Opisać wiadomości kontrolne.

1.2. Harmonogramowanie procesów

- Opisać algorytm Completely Fair Scheduler. [?]

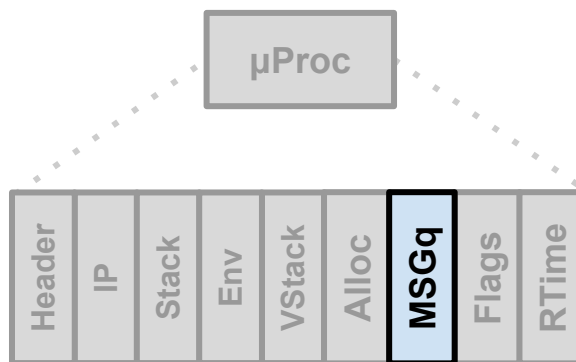


Rysunek 1.2: Schemat rejestrów wymaganych przez usprawnienia harmonogramowania SMP.

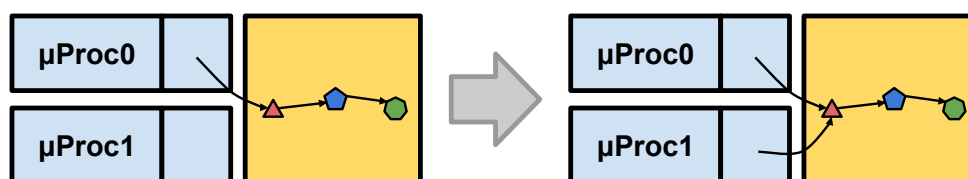
- Opisać implementację i wykorzystanie `RUNq`.
- Opisać implementację prymitywu `sleep` oraz `WAITq`.
- Opisać wady możliwe usprawnienia (load-balancing i dzielenie zużycia).

1.3. Implementacja Modelu Aktorowego

- Opisać powstawanie procesów i prymityw `spawn`.
- Opisać implementację prymitywów `send` oraz `recv`.
- Opisać logiczną autonomiczność procesów (brak mutacji = inne procesy nie mogą ingerować).
- Opisać sposób porozumiewania się procesów (kolejki nieblokujące). [?, ?]
- Opisać zmiany wprowadzone w stanie maszyny wirtualnej (dodatkowe rejestry).



Rysunek 1.3: Schemat rejestrów wymaganych przez implementację Modelu Aktorowego.



Rysunek 1.4: Schemat działania przesyłania wiadomości.

1.4. Implementacja przesyłania wiadomości

- Opisać implementację kolejek nieblokujących (+ weryfikacja poprawności). [?, ?]
- Opisać wykorzystanie CAS i problem ABA.
- Opisać krótko wady i możliwe usprawnienia zastosowanego rozwiązania (dynamic size, wait-free, optimistic FIFO). [?, ?, ?]
- Opisać krótko alternatywne podejścia (synchroniczne przekazywanie wiadomości - kanały, locki/mutexy/semafony).
- Opisać sposób pobierania wiadomości z kolejki i jego możliwe usprawnienia (pattern-matching).