

Implementacja maszyny wirtualnej dla funkcyjnych języków programowania wspierających przetwarzanie współbieżne.

Kajetan Rzepecki

**Wydział EAIiB
Katedra Informatyki Stosowanej**

24 stycznia 2014

Maszyna wirtualna - środowisko uruchomieniowe języków programowania uniezależniające je od platformy sprzętowej.

Maszyna wirtualna - środowisko uruchomieniowe języków programowania uniezależniające je od platformy sprzętowej.

Cele pracy:

- Implementacja interpretera kodu bajtowego.

Maszyna wirtualna - środowisko uruchomieniowe języków programowania uniezależniające je od platformy sprzętowej.

Cele pracy:

- ▶ Implementacja interpretera kodu bajtowego.
- ▶ Implementacja kolektora obiektów nieosiągalnych.

Maszyna wirtualna - środowisko uruchomieniowe języków programowania uniezależniające je od platformy sprzętowej.

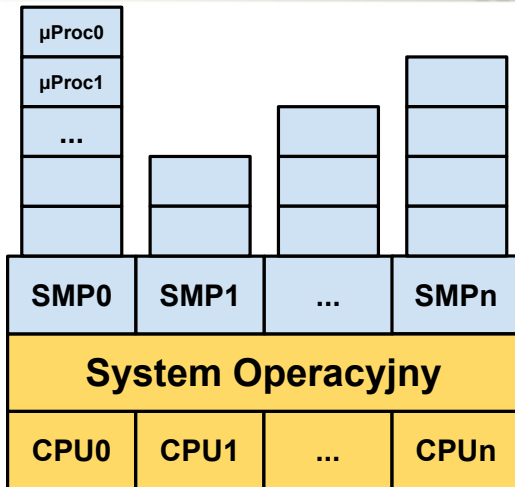
Cele pracy:

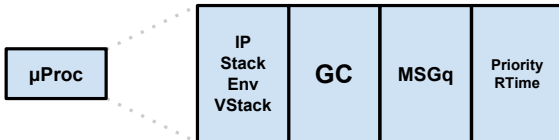
- ▶ Implementacja interpretera kodu bajtowego.
- ▶ Implementacja kolektora obiektów nieosiągalnych.
- ▶ Implementacja Modelu Aktorowego (ang. Actor Model).

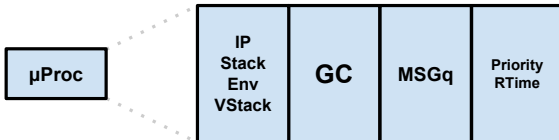
Maszyna wirtualna - środowisko uruchomieniowe języków programowania uniezależniające je od platformy sprzętowej.

Cele pracy:

- ▶ Implementacja interpretera kodu bajtowego.
- ▶ Implementacja kolektora obiektów nieosiągalnych.
- ▶ Implementacja Modelu Aktorowego (ang. Actor Model).
- ▶ Optymalizacja kosztownego kopiowania obiektów.

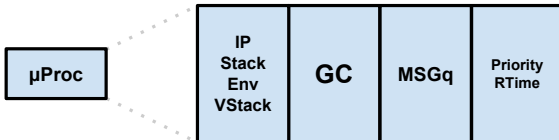






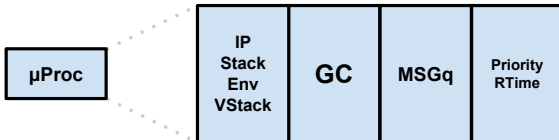
Oparty o **Three Instruction Machine**:

- Niewielka ilość rejestrów.



Oparty o **Three Instruction Machine**:

- ▶ Niewielka ilość rejestrów.
- ▶ Niewielka ilość instrukcji.



Oparty o **Three Instruction Machine**:

- ▶ Niewielka ilość rejestrów.
- ▶ Niewielka ilość instrukcji.
- ▶ Architektura **CISC**.

```
(define (add a b)  
  (primop + a b))
```

```
;; ...  
(add 2 2)
```

```
(define (add a b)
  (primop + a b))
```

```
;; ...
(add 2 2)
```

```
add:      TAKE
          TAKE
          NEXT __add0
          ENTER 1
__add0:   NEXT __add1
          ENTER 0
__add1:   PRIMOP '+'
          RETURN

# ...

__start:  NEXT 2
          NEXT 2
          ENTER add
```

- Celem jest uniknięcie nadmiernego kopiowania pamięci przy zachowaniu *logicznej* separacji procesów.

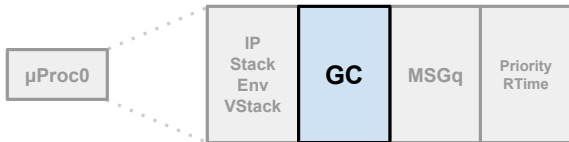
- ▶ Celem jest uniknięcie nadmiernego kopiowania pamięci przy zachowaniu *logicznej* separacji procesów.
- ▶ Pamięć jest współdzielona pomiędzy procesami, ale nie można jej modyfikować.

- ▶ Celem jest uniknięcie nadmiernego kopiowania pamięci przy zachowaniu *logicznej* separacji procesów.
- ▶ Pamięć jest współdzielona pomiędzy procesami, ale nie można jej modyfikować.
- ▶ Kolektor wykorzystuje *leniwe zliczanie referencji*.

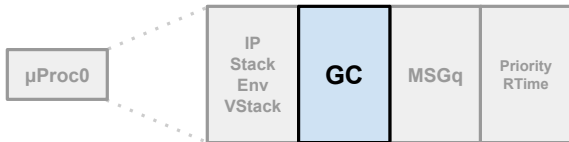
- ▶ Celem jest uniknięcie nadmiernego kopiowania pamięci przy zachowaniu *logicznej* separacji procesów.
- ▶ Pamięć jest współdzielona pomiędzy procesami, ale nie można jej modyfikować.
- ▶ Kolektor wykorzystuje *leniwe zliczanie referencji*.
- ▶ Kolekcja pamięci procesu nie zależy od innych procesów.

- ▶ Celem jest uniknięcie nadmiernego kopiowania pamięci przy zachowaniu *logicznej* separacji procesów.
- ▶ Pamięć jest współdzielona pomiędzy procesami, ale nie można jej modyfikować.
- ▶ Kolektor wykorzystuje *leniwe zliczanie referencji*.
- ▶ Kolekcja pamięci procesu nie zależy od innych procesów.
- ▶ “Ostatni gasi światło.”

- ▶ Celem jest uniknięcie nadmiernego kopiowania pamięci przy zachowaniu *logicznej* separacji procesów.
- ▶ Pamięć jest współdzielona pomiędzy procesami, ale nie można jej modyfikować.
- ▶ Kolektor wykorzystuje *leniwe zliczanie referencji*.
- ▶ Kolekcja pamięci procesu nie zależy od innych procesów.
- ▶ “Ostatni gasi światło.”
- ▶ Proste obiekty (≤ 8 bajtów) są kopiowane.

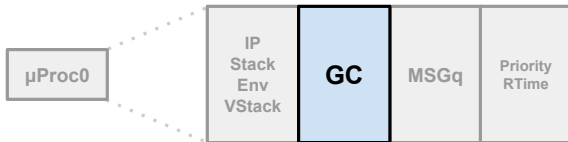


Polega na opóźnieniu kolekcji obiektu do następnej alokacji poprzez wykorzystanie listy zwolnionych obiektów.



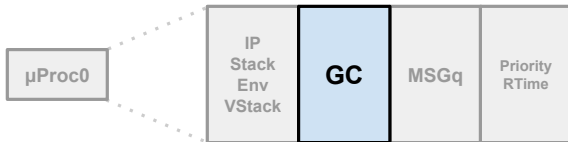
Polega na opóźnieniu kolekcji obiektu do następnej alokacji poprzez wykorzystanie listy zwolnionych obiektów.

- Szybkie dealokacje.



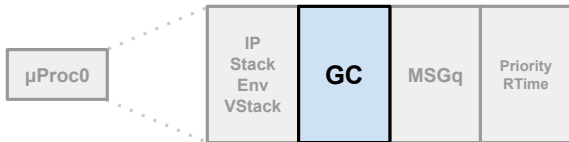
Polega na opóźnieniu kolekcji obiektu do następnej alokacji poprzez wykorzystanie listy zwolnionych obiektów.

- ▶ Szybkie dealokacje.
- ▶ Szybkie alokacje zamortyzowane listą wolnych obiektów.



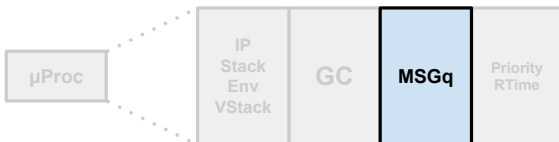
Polega na opóźnieniu kolekcji obiektu do następnej alokacji poprzez wykorzystanie listy zwolnionych obiektów.

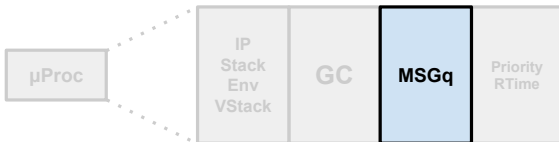
- ▶ Szybkie dealokacje.
- ▶ Szybkie alokacje zamortyzowane listą wolnych obiektów.
- ▶ Pamięć nie jest natychmiastowo zwracana do Systemu Operacyjnego.



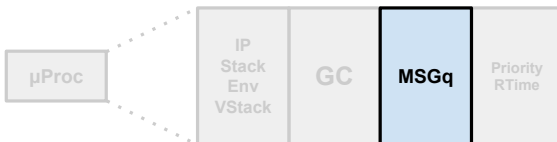
Polega na opóźnieniu kolekcji obiektu do następnej alokacji poprzez wykorzystanie listy zwolnionych obiektów.

- ▶ Szybkie dealokacje.
- ▶ Szybkie alokacje zamortyzowane listą wolnych obiektów.
- ▶ Pamięć nie jest natychmiastowo zwracana do Systemu Operacyjnego.
- ▶ Wymaga atomowych operacji na liczniku referencji oraz barier pamięci.

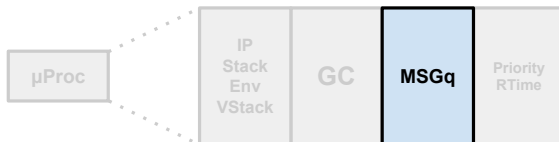




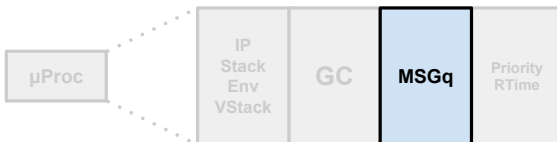
- Procesy są obiektami “pierwszej klasy”.



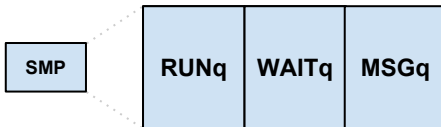
- ▶ Procesy są obiektami “pierwszej klasy”.
- ▶ Identyfikator procesu (pid) to wskaźnik na kontekst procesu.

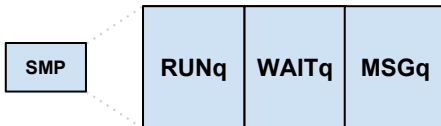


- ▶ Procesy są obiektami “pierwszej klasy”.
- ▶ Identyfikator procesu (pid) to wskaźnik na kontekst procesu.
- ▶ Wiadomości są przesyłane asynchronicznie.

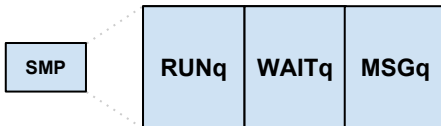


- ▶ Procesy są obiektami “pierwszej klasy”.
- ▶ Identyfikator procesu (pid) to wskaźnik na kontekst procesu.
- ▶ Wiadomości są przesyłane asynchronicznie.
- ▶ Implementacja wykorzystuje kolejki nieblokujące.

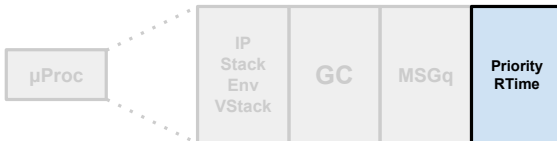




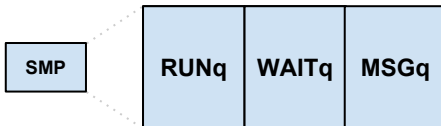
- Implementacja wykorzystuje Model Aktorowy!



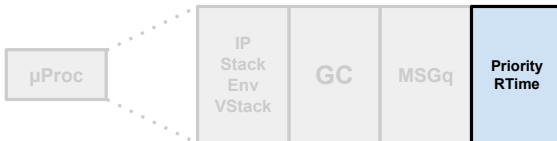
- Implementacja wykorzystuje Model Aktorowy!



- Procesy są wywłaszczane (ang. preemptive concurrency).



- Implementacja wykorzystuje Model Aktorowy!



- Procesy są wywłaszczane (ang. preemptive concurrency).
- Wykorzystuje algorytm **Completely Fair Scheduling**.

- ▶ Udało się osiągnąć zamierzone cele.

- ▶ Udało się osiągnąć zamierzone cele.
- ▶ Nie udało się zawrzeć wszystkich zaplanowanych optymalizacji.

- ▶ Udało się osiągnąć zamierzone cele.
- ▶ Nie udało się zawrzeć wszystkich zaplanowanych optymalizacji.
- ▶ Udało się zidentyfikować wiele kierunków przyszłego rozwoju.