Dominik Wróbel		
Projekt myjnia samochodowa w języku Erlang		
Rok III	Informatyka	Grupa Środa 09:30
Data oddania projektu:	18 grudnia 2019	

Spis treści

1 Cel programu

Celem programu jest symulacja myjni samochodowej z zastosowaniem podejścia wielowątkowego. Rozważana myjnia spełnia założenia:

- W danym momencie czasu tylko jeden samochód może być w myjni
- Myjnia nalicza opłaty w systemie czasowym, po wjechaniu do myjni od momentu rozpoczęcia użytkowania urządzeń myjących, aż do zakończenia mycia
- Sekunda mycia samochodu w myjni kosztuje 50 groszy
- Po zakończeniu mycia myjnia zwraca kierowcy informację o naliczonej opłacie

2 Opis programu

2.1. Wątki

W programie osobne wątki stanowią:

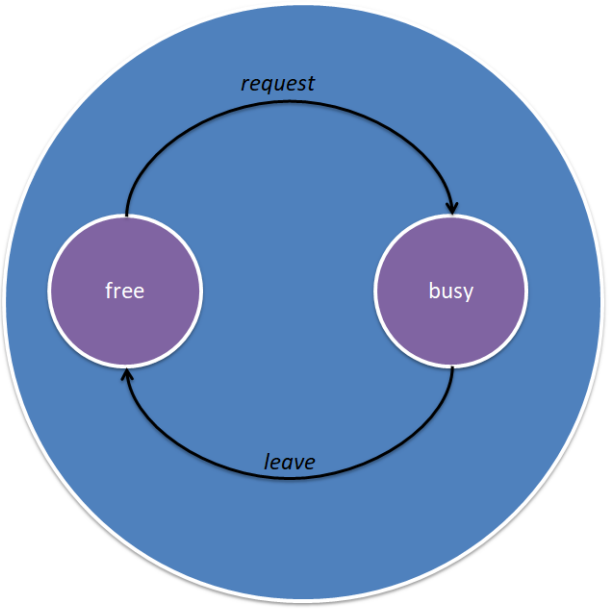
- Myjnia - to wątek działający w pętli, który obsługuje przyjeżdżające samochody. Obsługa samochodów została zaimplementowana przy użyciu semafora binarnego, tak aby w danej chwili czasu tylko jeden samochód był w myjni oraz aby samochody który przyjechały pierwsze były wcześniej w kolejce (FIFO)

- Samochód - każdy z odwiedzających myjnie samochodów to osobny wątek, wątki te komunikują się z myjnią w celu uzyskania dostępu do myjni oraz wykonania operacji związanych z myciem i płatnością
- Timer - to element myjni, który służy do zliczania czasu mycia dla pojedynczego samochodu, timer został zaimplementowany jako osobny wątek

2.2. Schematy działania i komunikacji wątków

2.2.1. Myjnia - obsługa wielu wątków przez semafor

Semafor binarny jest początkowo w stanie *free*, kiedy otrzyma wiadomość *request* (żądanie o dostęp do myjni), przechodzi w stan *busy*. Wówczas pozostały wiadomości *request* są kolejkowane i czekają aż semafor znów będzie stanie *free*.



Rysunek 1: Schemat działania semafora binarnego.

Listing 1: Fragmenty kodu realizujace semafo - zamykanie semafora

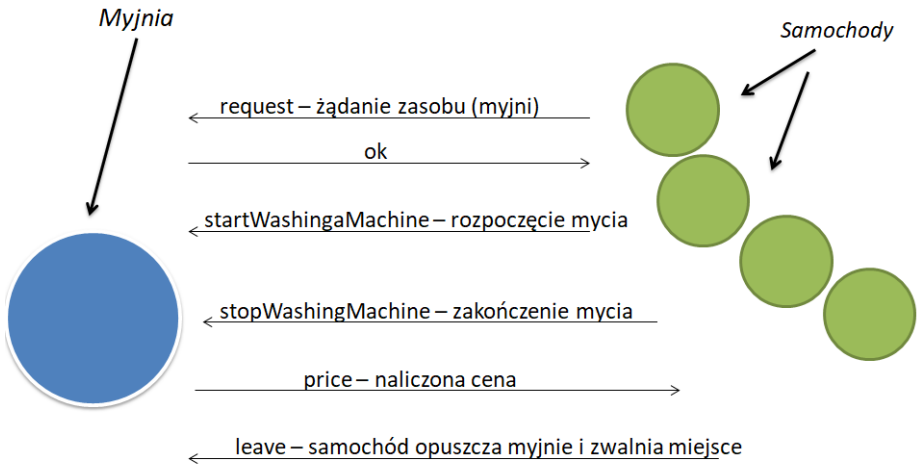
```
% car wash is free initially
init() ->
    free().

% car wash waits for clients to arrive...
free() ->
    receive
        {request, Pid} ->   % give driver access to car wash
            io:format("CAR-WASH Car with ~p got a car wash. ~n", [Pid]),
            Pid ! ok,       % send back approval
            busy(Pid);      % car wash is busy now...
        close ->
            close()         % close car wash
    end.
```

Listing 2: Fragmenty kodu realizujace semafor - otwieranie semafora

```
busy(Pid) ->
  io:format("CAR-WASH MY PID IS ~p. ~n", [self()]),
    receive
        {leave, Pid} ->
            free(); % open the semaphore
      startWashingMachine ->
        io:format("CAR-WASH Car with ~p started timer. ~n", [Pid]),
        startTimer(),
        busy(Pid);
      endWashingMachine ->
        io:format("CAR-WASH Car with ~p ended timer. ~n", [Pid]),
        endTimer(),
        busy(Pid);
      {timestamp, Val} ->
        io:format("CAR-WASH Timer sent me value ~p. ~n", [Val]),
        Pid ! {price, Val*2},
        busy(Pid)
    end.
```

2.2.2. Komunikacja myjnia - samochód

Samochód wysyła do myjni żądania o dostęp do myjni, następnie rozpoczyna mycie, a po jego zakończeniu dostaje informacje o naliczonej cenie.



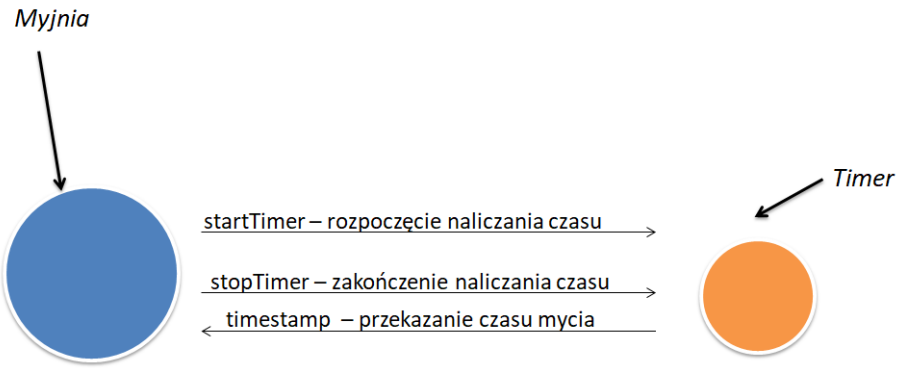
Rysunek 2: Komunikacja myjnia - samochód.

Listing 3: Fragmenty kodu realizujace komunikacje samochód - myjnia

```
% carDriver sends request to get to the car wash
requestCarWash() ->
  io:format("CAR with pid ~p request a car wash !~n", [self()]),
    CarWashPid = whereis(carWashMutex),
  CarWashPid ! {request, self()},
    receive
    ok ->
      io:format("CAR car ~p starts washing machine... ~n", [self()]),
      startWashingMachine(),
      washCar()
  end.

washCar() ->
  io:format("CAR car ~p is washing... ~n", [self()]),
    I = random:uniform(2000) + 3000,
  timer:sleep(I),
  stopWashingMachine(),
  receive
    {price, Value} ->
      io:format("CAR I got price ~p ~n", [Value]),
      leaveCarWash()
  end.
```

2.2.3. Komunikacja myjnia - timer

Myjnia komunikuje się z Timerem w celu uzyskania informacji o czasie mycia samochodu. Na podstawie tego czasu oblicza cenę, którą przekazuje kierowcy.



Rysunek 3: Komunikacja myjnia - timer.

Listing 4: Fragmenty kodu realizujace komunikacje samochód - timer

```
% car wash
busy(Pid) ->
  io:format("CAR-WASH MY PID IS ~p. ~n", [self()]),
    receive
        {leave, Pid} ->
            free();
    startWashingMachine ->
      io:format("CAR-WASH Car with ~p started timer. ~n", [Pid]),
      startTimer(),
      busy(Pid);
    endWashingMachine ->
      io:format("CAR-WASH Car with ~p ended timer. ~n", [Pid]),
      endTimer(),
      busy(Pid);
    {timestamp, Val} ->
      io:format("CAR-WASH Timer sent me value ~p. ~n", [Val]),
      Pid ! {price, Val*2},
      busy(Pid)
    end.

% timer
initTimer() ->
  startListening(0).

startListening(StartTime) ->
  receive
    start ->
        % io:format("TIMER started... ~n"),
      startListening(os:system_time());
    finish ->
      % io:format("TIMER finished... ~n"),
      carWashMutex ! {timestamp, (os:system_time()-StartTime)/1000},
      startListening(0)
    end.
```

3 Użyte funkcje z innych modułów

- Funkcja *os:system_time()* - w celu obliczania czasu trwania mycia samochodu na podstawie czasu systemowego
- Funkcja *timer:sleep()* - w celu symulacji dłuższego czasu mycia samochodu
- Funkcja *random:uniform()* - w celu wylosowania czasu mycia samochodu

4 Rozwiązania problemów

- Symulacja czasu mycia samochodu została zaimplementowana jako wartość losowa przy użyciu funkcji opóźniającej czas wykonania
- Czas mycia samochodu obliczany jest jako różnica czasu systemowego od pomiędzy momentem otrzymania wiadomości o zakończeniu i starcie mycia

5 Instrukcja obsługi

W celu uruchomienia programu dla czterech samochodów należy uruchomić shell Erlanga w miejscu gdzie znajdują się pliki i wpisać:

Listing 5: Uruchomienie programu

```
c(carWashMutex).
c(carDriver).
c(myTimer).
carWashMutex:startCarWash().
myTimer:createTimer().
carDriver:getNewCarsToWash(). % creates four cars
```

6 Przykładowy test

Przykładowy test programu dla czterech samochodów wyświetla logi informujące o stanie każdego z wątków:

Listing 6: Przykładowe uruchomienie

```
7> carDriver:getNewCarsToWash().
CAR with pid <0.99.0> comes to car wash !
CAR with pid <0.99.0> request a car wash !
CAR with pid <0.100.0> comes to car wash !
CAR with pid <0.100.0> request a car wash !
CAR-WASH Car with <0.99.0> got a car wash.
CAR with pid <0.101.0> comes to car wash !
CAR with pid <0.101.0> request a car wash !
CAR-WASH MY PID IS <0.95.0>.
CAR car <0.99.0> starts washing machine...
CAR with pid <0.102.0> comes to car wash !
CAR with pid <0.102.0> request a car wash !
CAR with pid <0.99.0> starts a car wash !
CAR car <0.99.0> is washing...
CAR-WASH Car with <0.99.0> started timer.
```

```
ok
CAR-WASH MY PID IS <0.95.0>.
8>
CAR with pid <0.99.0> stops a car wash !
CAR-WASH Car with <0.99.0> ended timer.
CAR-WASH MY PID IS <0.95.0>.
CAR-WASH Timer sent me value 4003.84.
CAR-WASH MY PID IS <0.95.0>.
CAR I got price 8007.68
CAR with pid <0.99.0> leaves a car wash !
CAR-WASH Car with <0.100.0> got a car wash.
CAR-WASH MY PID IS <0.95.0>.
CAR car <0.100.0> starts washing machine...
CAR with pid <0.100.0> starts a car wash !
CAR car <0.100.0> is washing...
CAR-WASH Car with <0.100.0> started timer.
CAR-WASH MY PID IS <0.95.0>.
CAR with pid <0.100.0> stops a car wash !
CAR-WASH Car with <0.100.0> ended timer.
CAR-WASH MY PID IS <0.95.0>.
CAR-WASH Timer sent me value 3979.264.
CAR-WASH MY PID IS <0.95.0>.
CAR I got price 7958.528
CAR with pid <0.100.0> leaves a car wash !
CAR-WASH Car with <0.101.0> got a car wash.
CAR-WASH MY PID IS <0.95.0>.
CAR car <0.101.0> starts washing machine...
CAR with pid <0.101.0> starts a car wash !
CAR car <0.101.0> is washing...
CAR-WASH Car with <0.101.0> started timer.
CAR-WASH MY PID IS <0.95.0>.
CAR with pid <0.101.0> stops a car wash !
CAR-WASH Car with <0.101.0> ended timer.
CAR-WASH MY PID IS <0.95.0>.
CAR-WASH Timer sent me value 3988.48.
CAR-WASH MY PID IS <0.95.0>.
CAR I got price 7976.96
CAR with pid <0.101.0> leaves a car wash !
CAR-WASH Car with <0.102.0> got a car wash.
CAR-WASH MY PID IS <0.95.0>.
CAR car <0.102.0> starts washing machine...
CAR with pid <0.102.0> starts a car wash !
CAR car <0.102.0> is washing...
CAR-WASH Car with <0.102.0> started timer.
CAR-WASH MY PID IS <0.95.0>.
CAR with pid <0.102.0> stops a car wash !
CAR-WASH Car with <0.102.0> ended timer.
CAR-WASH MY PID IS <0.95.0>.
CAR-WASH Timer sent me value 3984.384.
CAR-WASH MY PID IS <0.95.0>.
CAR I got price 7968.768
CAR with pid <0.102.0> leaves a car wash !
```

7 Możliwe rozszerzenia programu

Program można rozszerzyć np. poprzez dodanie bardziej złożonej logiki mycia samochodu, a także poprzez dodanie drugiego stanowiska do mycia samochodów.