Pracownia z Kursu języka Erlang
Lista 4

1. **(3pkt)** Celem tego zadania jest implementacja systemu mierzenia wydajności tworzenia procesów erlangowych i przesyłania komunikatów między nimi.

   Moduł `ring` odpowiada za tworzenie jednokierunkowego pierścienia procesów. Powinien implementować jedną funkcję `start`, która powinna przyjmować jako argumenty `pid` procesu nadzorcy, `pid` procesu statystyk i długość pierścienia. Przed rozpoczęciem konstruowania pierścienia należy wysłać do procesu statystyk komunikat `{ring,start,RingId,Length,CurrentTime}`. Po utworzeniu należy wysłać do procesu nadzorcy komunikat `{ring,created,RingId}`, a do procesu statystyk `{ring,stop,RingId,CurrentTime}`.

   Procesy w pierścieniu powinny:

   - znać `pid` procesu statystyk, nadzorcy oraz swojego prawego sąsiada,
   - być ponumerowane od 0 do N,
   - umieć przyjmować żetony postaci w postaci `{TokenId,Steps,StopTime}`, które przekażą dalej po zmniejszeniu wartości `Steps` o jeden,
   - wysłać do procesu statystyk komunikat, jeśli `Steps` jest równe 0 lub `StopTime` zostało przekroczone; komunikat ma być postaci `{token,stop,TokenId,Steps,ProcessNum,CurrentTime}`,
   - po otrzymaniu komunikatu `stop`, mają przekazać go dalej i zakończyć swoje działanie.

   Jeśli pola żetonu `Steps` albo `StopTime` są atomem `undefined` to mają być ignorowane w obliczeniach.

   Po zniszczeniu pierścienia należy wysłać komunikat `{ring,destroyed,RingId}`.

   Do procesu statystyk należy wysłać komunikat `{token,start,TokenId,Steps,CurrentTime}` przed przekazaniem nowego żetonu do pierwszego procesu w pierścieniu.

   Moduł `statistics` powinien zawierać proces kolekcjonujący statystyki. Funkcja `start` powinna go tworzyć i zwracać jego `pid`. Proces ten powinien:

   - zarządzać informacjami o żetonach `record(token,{StartTime,StopTime,Steps,StepsDone,ProcessNum})`,
   - zarządzać statystykami o pierścieniach `record(ring,{StartTime,StopTime,Length})`,
   - po otrzymaniu komunikatu `{Atom,report}` wydrukować jakąś ciekawą interpretację zebranych statystyk (zostawiam Wam dowolność),
   - po otrzymaniu komunikatu `{Atom,reset}` wymazać odpowiednio wszystkie informacje o żetonach lub pierścieniach,
   - zakończyć się po otrzymaniu komunikatu `stop`.

   Funkcjonalność procesu nadzorcy zbierz w module `benchmark`. Przygotuj testy, które pokażą ile (średnio) procesów można utworzyć na sekundę, ile komunikatów można przekazać na sekundę. Jeśli masz dostęp do wieloprocesorowej maszyny zobacz jak zachowuje się na niej maszyna wirtualna Erlanga. Zobacz czy uda Ci się utworzyć milion i więcej procesów lub komunikatów. Wykonaj też test z małym pierścieniem i mnóstwem komunikatów – zobacz czy po danym czasie komunikaty w miarę równomiernie rozłożą się między procesami.

   Do implementacji zadania mogą się przydać moduły `dict` i `timer` oraz referencje.

Lista i materiały znajdują się pod adresem
http://cahirwpz.cs.uni.wroc.pl/main-pl/erlang-language-summer-2010/ *Krystian Bacławski*