Erlang

Erlang Top

Erlang

Erlang Top (etop) jest narzędziem do prezentowania informacji dotyczących procesów Erlanga.

Wynik jego działania podobny jest do informacji prezentowanych przez UNIX-owy *top*.

Licencja, dostęp

-

- Etop znajduje się w bibliotece *observer*

Przykładowy widok

```
==============================================================================
 tiger@durin                                                        13:40:32
 Load:  cpu         0               Memory:  total        1997    binary         33
        procs     197                        processes       0    code          173
        runq      135                        atom         1002    ets            95

Pid            Name or Initial Func    Time    Reds  Memory    MsgQ Current Function
------------------------------------------------------------------------------
<127.23.0>     code_server                0   59585   78064       0 gen_server:loop/6
<127.21.0>     file_server_2              0   36380   44276       0 gen_server:loop/6
<127.2.0>      erl_prim_loader            0   27962    3740       0 erl_prim_loader:loop
<127.9.0>      kernel_sup                 0    6998    4676       0 gen_server:loop/6
<127.17.0>     net_kernel                62    6018    3136       0 gen_server:loop/6
<127.0.0>      init                       0    4156    4352       0 init:loop/1
<127.16.0>     auth                       0    1765    1264       0 gen_server:loop/6
<127.18.0>     inet_tcp_dist:accept       0     660    1416       0 prim_inet:accept0/2
<127.5.0>      application_controll       0     569    6756       0 gen_server:loop/6
<127.137.0>    net_kernel:do_spawn_       0     553    5840       0 dbg:do_relay_1/1
==============================================================================
```

Opis

tiger@durin - nazwa węzła, którego procesy obserwujemy

Load:

- ***cpu*** – procent czasu, kiedy węzeł był aktywny **,**
- ***procs*** – liczba procesów danego węzła,
- ***runq*** – liczba procesów, które są gotowe do uruchomienia.

```
==============================================================================
tiger@durin                                                        13:40:32
Load:  cpu         0               Memory:  total        1997    binary         33
       procs     197                        processes       0    code          173
       runq      135                        atom         1002    ets            95

Pid            Name or Initial Func    Time    Reds  Memory    MsgQ Current Function
------------------------------------------------------------------------------
<127.23.0>     code_server                0   59585   78064       0 gen_server:loop/6
<127.21.0>     file_server_2              0   36380   44276       0 gen_server:loop/6
<127.2.0>      erl_prim_loader            0   27962    3740       0 erl_prim_loader:loop
<127.9.0>      kernel_sup                 0    6998    4676       0 gen_server:loop/6
<127.17.0>     net_kernel                62    6018    3136       0 gen_server:loop/6
<127.0.0>      init                       0    4156    4352       0 init:loop/1
<127.16.0>     auth                       0    1765    1264       0 gen_server:loop/6
<127.18.0>     inet_tcp_dist:accept       0     660    1416       0 prim_inet:accept0/2
<127.5.0>      application_controll       0     569    6756       0 gen_server:loop/6
<127.137.0>    net_kernel:do_spawn_       0     553    5840       0 dbg:do_relay_1/1
==============================================================================
```

Opis

13:40:32 **–** aktualny czas [hh:mm:ss]

Memory – pamięć zaalokowana przez węzeł w kilobajtach

```
==========================================================================================
 tiger@durin                                                          13:40:32
 Load:   cpu         0                Memory:  total         1997    binary         33
         procs     197                         processes        0    code          173
         runq      135                         atom          1002    ets            95

Pid            Name or Initial Func    Time    Reds  Memory    MsgQ Current Function
------------------------------------------------------------------------------------------
<127.23.0>     code_server                0   59585   78064       0 gen_server:loop/6
<127.21.0>     file_server_2              0   36380   44276       0 gen_server:loop/6
<127.2.0>      erl_prim_loader            0   27962    3740       0 erl_prim_loader:loop
<127.9.0>      kernel_sup                 0    6998    4676       0 gen_server:loop/6
<127.17.0>     net_kernel                62    6018    3136       0 gen_server:loop/6
<127.0.0>      init                       0    4156    4352       0 init:loop/1
<127.16.0>     auth                       0    1765    1264       0 gen_server:loop/6
<127.18.0>     inet_tcp_dist:accept       0     660    1416       0 prim_inet:accept0/2
<127.5.0>      application_controll       0     569    6756       0 gen_server:loop/6
<127.137.0>    net_kernel:do_spawn_       0     553    5840       0 dbg:do_relay_1/1
==========================================================================================
```

Opis

Dla każdego procesu wyświetlane są następujące informacje:

- ***Pid*** – identyfikator procesu,
- ***Name...*** – nazwa procesu,
- ***Time*** – czas działania procesu,
- ***Reds*** – liczba redukcji (każda operacja w systemie np. wywołanie funkcji w pętli, BIF-y itp.) wykonanych w procesie,

```
==============================================================================
 tiger@durin                                                        13:40:32
 Load:  cpu         0               Memory:  total        1997    binary         33
        procs     197                        processes       0    code          173
        runq      135                        atom         1002    ets            95

Pid            Name or Initial Func    Time    Reds  Memory    MsgQ Current Function
------------------------------------------------------------------------------
<127.23.0>     code_server                0   59585   78064       0 gen_server:loop/6
<127.21.0>     file_server_2              0   36380   44276       0 gen_server:loop/6
<127.2.0>      erl_prim_loader            0   27962    3740       0 erl_prim_loader:loop
<127.9.0>      kernel_sup                 0    6998    4676       0 gen_server:loop/6
<127.17.0>     net_kernel                62    6018    3136       0 gen_server:loop/6
<127.0.0>      init                       0    4156    4352       0 init:loop/1
<127.16.0>     auth                       0    1765    1264       0 gen_server:loop/6
<127.18.0>     inet_tcp_dist:accept       0     660    1416       0 prim_inet:accept0/2
<127.5.0>      application_controll       0     569    6756       0 gen_server:loop/6
<127.137.0>    net_kernel:do_spawn_       0     553    5840       0 dbg:do_relay_1/1
==============================================================================
```

Opis

- ***Memory*** – rozmiar procesu w bajtach, otrzymywany przez wywołanie *process_info(pid, memory),*
- ***MsgQ*** – (message queue) długość kolejki wiadomości procesu,
- ***Current Function*** – obecnie wywołana funkcja.

```
==============================================================================
 tiger@durin                                                        13:40:32
 Load:  cpu         0               Memory:  total        1997    binary         33
        procs     197                        processes       0    code          173
        runq      135                        atom         1002    ets            95

Pid            Name or Initial Func    Time    Reds  Memory    MsgQ Current Function
------------------------------------------------------------------------------
<127.23.0>     code_server                0   59585   78064       0 gen_server:loop/6
<127.21.0>     file_server_2              0   36380   44276       0 gen_server:loop/6
<127.2.0>      erl_prim_loader            0   27962    3740       0 erl_prim_loader:loop
<127.9.0>      kernel_sup                 0    6998    4676       0 gen_server:loop/6
<127.17.0>     net_kernel                62    6018    3136       0 gen_server:loop/6
<127.0.0>      init                       0    4156    4352       0 init:loop/1
<127.16.0>     auth                       0    1765    1264       0 gen_server:loop/6
<127.18.0>     inet_tcp_dist:accept       0     660    1416       0 prim_inet:accept0/2
<127.5.0>      application_controll       0     569    6756       0 gen_server:loop/6
<127.137.0>    net_kernel:do_spawn_       0     553    5840       0 dbg:do_relay_1/1
==============================================================================
```

Uruchamianie

Narzędzie **etop** powinniśmy uruchamiać za pomocą odpowiedniego skryptu:

Windows:


```
1  @ECHO OFF
2  CALL werl -sname etop -hidden -s etop -s erlang halt -output text  %*
```

UNIX:


```
1  #!/bin/sh
2
3  NAME="etop"
4  erl -sname $NAME -hidden -s etop -s erlang halt -output text $@
```

Uruchamianie

Węzeł, który chcemy monitorować należy podać jako parametr (atom) w konsoli w formacie:

-node nazwa@host

Przykładowy skrypt może wyglądać następująco:

```
1 @ECHO OFF
2 CALL werl -sname etop
3           -hidden
4           -s etop start -node queue@TOMASZ-TOSHIBA
5           -s etop config -interval 10
6           -s etop config -sort msg_q
7           -s erlang halt -output text %*
```

Uruchamianie

Dostępne są ponadto opcjonalne parametry uruchamiania:

- ***-setcookie cookieName***
  Ciasteczko dla węzła *etop*. Musi być takie samo jak dla węzła monitorowanego.
  Argument: atom.
- ***-lines 10***
  Ilość linii (procesów) do wyświetlenia.
  Argument: integer.
- ***-interval 10***
  Odstęp w sekundach między kolejnymi wyświetleniami.
  Argument: integer.

Uruchamianie

- ***-accumulate true/false***
  Określa jak prezentowane są nagłówki **Time** oraz **Reds**. Domyślnie false.
  true → od ostatniego wyświetlenia
  false → od początku
  Argument: atom (true lub false).
- ***-sort memory***
  Ustawienia sortowania. Domyślnie runtime (jeżeli tracing=off to reductions).
  Dostępne argumenty (atomy): runtime | reductions | memory | msg_q
- ***-tracing on/off***
  Etop używa erlangowskiej funkcji śledzenia, dlatego nie ma innej możliwości monitorowania węzła, dopóki funkcja *tracing* jest uruchomiona.
  Dostępne argumenty: on lub off.

Funkcje

Możemy użyć także następujących funkcji z modułu etop:

- ***start() → ok***
  Uruchomienie narzędzia *etop* (preferuje się jednak użycie skryptu).
- ***start([Options]) → ok***
  Uruchomienie z dodatkowymi parametrami. Opcje podajemy jako krotki 2-atomowe. Możliwe jest zastosowanie takich parametrów, jak te startowe wpisywane w konsolę (wyjątek: cookie jako string) oraz dodatkowo *ustawienie portu: {port, integer}.*
- ***config(Key, Value) → Result***
  Zmiana parametrów w czasie działania narzędzia.
- ***dump(File) → Result***
  Zapis wyświetlanych danych do pliku podanego jako argument (string).
- ***stop() → stop***
  Zatrzymanie narzędzia etop.