

TAGP

Paul.Valckenaers@kuleuven.be

<https://www.linkedin.com/in/takashoo/>

[https://www.researchgate.net/profile/Paul Valckenaers](https://www.researchgate.net/profile/Paul_Valckenaers)

TAGP

- Onderzoeksgebaseerd >> extreem betrouwbare/beschikbare s/w
- Anders dan de meeste vakken tot nu toe
 - Meer vrijheid, minder zekerheden
 - Meer zelfstandig werken, beslissen en zelf kennis verwerven
 - ...
- Basis volstaat om te slagen → geen ongezonde stress
 - Ernstige inspanning en op tijd de nodige **vragen stellen!**
 - De rest is bonus; eigen schuld als je niet genoten hebt.

Erlang – Spawned Shelter!

Erlang is a programming language that was designed to create reliable, evolvable and low latency systems.

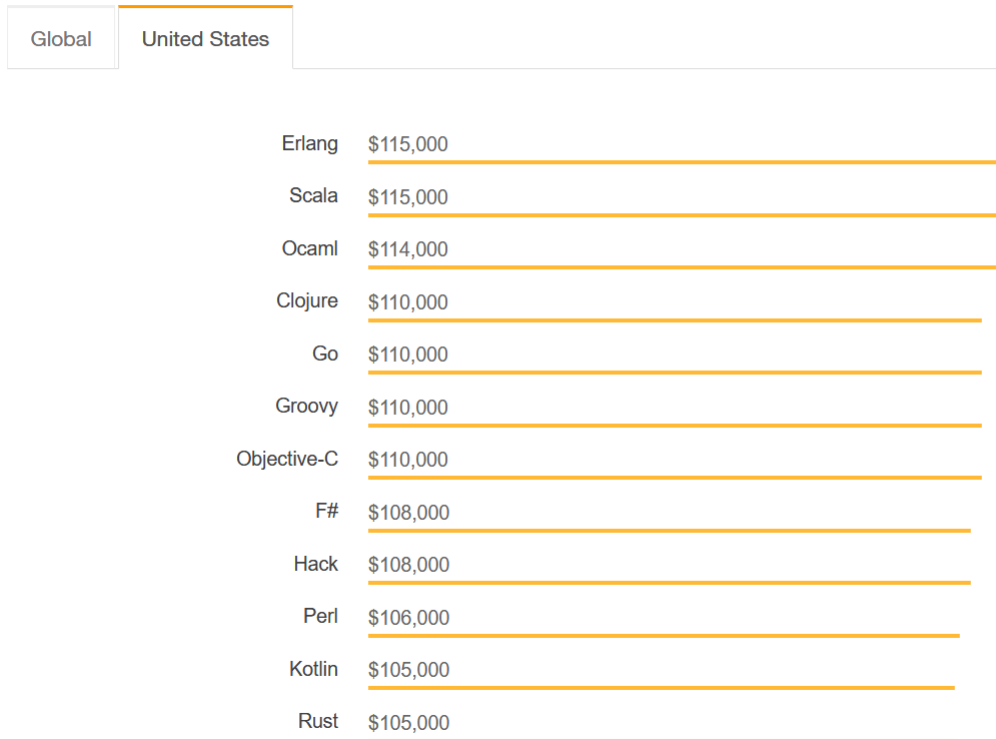
It facilitates the job of engineers building fault tolerant and maintainable systems.

That is why, in contrast to most programming languages, fault tolerance, concurrency, distribution and tracing **were built into the core language from the beginning.**

Best-betaalde software ontwikkelaars in USA in 2017.

<https://insights.stackoverflow.com/survey/2018#top-paying-technologies>

What Languages Are Associated with the Highest Salaries Worldwide?



Erlang : hoge beschikbaarheid, ...

- Een eco-system (cf. <http://spawnedshelter.com>)
 - ERTS
 - BEAM
 - Erlang
 - OTP
 - WhatsApp
 - Web server: Cowboy
 - Embedded: <https://www.grisp.org/specs.html>
 - Elixir
 - Phoenix
 - Nerves
 - Python syntax, look-and-feel, ...

MOOC & on-line tutorials – Erlang

- <https://www.cs.kent.ac.uk/ErlangMasterClasses/#> (on youtube).
 - <https://www.futurelearn.com/courses/functional-programming-erlang>
 - <https://www.futurelearn.com/courses/concurrent-programming-erlang>
- Udemy MOOC
 - <https://www.udemy.com/master-erlang-programming-in-just-4-hours/>
 - <https://www.udemy.com/erlang-programming-for-beginners/>
- Tutorials & on-line books
 - <http://www.tryerlang.org/>
 - <https://www.ibm.com/developerworks/library/os-erlang1/index.html>
 - <https://www.tutorialspoint.com/erlang/>
 - <https://learnyoussomeerlang.com/contents>

Optie 1: Elektronica is jouw ding

- a. Raspberry Pi (iets zoals een Pi zero of <https://www.grisp.org/specs.html>)

Verbonden aan ...

- b. Erlang en C (port) en/of Erlang ALE (evt. vergelijken)
http://opensource.erlang-solutions.com/erlang_ale/
https://github.com/esl/erlang_ale

- c. BONUS: Monitoring, supervisory tree, UPS, ...

>> In groep, voeding aan/uit zetten, netwerk onderbreken...

ofwel

- c. BONUS: SeL4 en Erlang (voor de hackers), microSD en USB ontzien, ...

Optie 1: bit-niveau

Subject: KNX Converter

MSB 0x1A , LSB 0x7A >> 50.72 °C

-module(convertierKNXtemperatuur).

-export([convert/1]).

% eerst omzetten in een bitstring

convert([MSB, LSB]) -> convert2(<<MSB:8, LSB:8>>).

% when sign bit equals zero:

convert2(<<0:1, E:4, M:11>>) -> math:pow(2, E) * M / 100;

%when sign bit equals one:

convert2(<<1:1, E:4, M:11>>) -> math:pow(2, E) * (M - 2048) / 100.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace KNXDemo.KNX
{
    class KNXConverter
    {
        public KNXConverter()
        {

        }

        public static double TempFromEIS5(byte[] data)
        {
            // EIS 5 - DPT 9 Format: 2-Octet Float Value (DPT 9.001 Temperature [°C])
            // MSB: MEEE EMMMM
            // LSB: MMMM MMMM
            // Encoding: (0.01*M)*2^E, M is in 2's complement

            bool complement = (((data[0] & 0x00000080) >> 7) == 0x00000001);
            uint exp = (uint)((data[0] & 0x00000078) >> 3);
            uint m = (uint)((data[0] & 0x00000007) << 8 | data[1]);
            double mantisse = 0;

            // 2's complement check
            if (!complement)
            {
                mantisse = m;
            }
            else
            {
                m = m - 1;
                m = ~m;
                m = m & 0x0000007FF;
                mantisse = m * (-1);
            }

            double result = (0.01d * mantisse) * Math.Pow(2, exp);
            return Math.Round(result, 2);
        }
    }
}
```


Optie 2: Web servers zijn jouw ding

- Cowboy en
 - een front end met wat gekend is, jouw voorkeur geniet, ...
 - > (kijk evt. eens naar) <http://zotonic.com/page/520/erlydtl>
 - of
 - met ELM
 - > een functionele taal die gebruiksgemak als prioriteit heeft
 - > ELM <http://elm-lang.org/>
- BONUS: Show case voor “*Demo 1 in STORY*”

Optie 3: Python liefhebbers

- Erlang syntax is een echte bron van frustratie:
 - Elixir en Phoenix (web server)
 - BONUS - STORY show case.

Ofwel

- Automated conversion Erlang – Elixir (evt. ook efene behandelen).
- BONUS – toepassen op STORY case.

Optie 4: Testen en verificatie

- Meest nuttige optie voor de latere loopbaanmogelijkheden
 - a. Unit test, Common test.
 - b. Dialyzer.
 - c. PropEr, QuickCheck (lite).
 - d. DevOps (rebar3, ...).

Optie 5: Als je de juiste person kent

- Embedded, networked digital/virtual twin voor slimme energie-toepassingen

>> warmtepomp, PV, e-vehicle, ...

Erlang programmeertaal

- <http://www.tryerlang.org/>
- http://erlang.org/doc/getting_started/users_guide.html
- <https://www.tutorialspoint.com/erlang/>
- <https://learnyoussomeerlang.com/contents>

Erlang – wat is er anders?

- Pattern matching (IMHO verslavend)
- Alle variabelen zijn ‘final’
- Geen for, while, ... maar *recursie*
- Staartrecursie en accumulerende parameter(s).
- Geen new maar spawn.

Wordt vervolgt...