Software Technology

Josef Svenningsson

Software Technology

Software Technology

Området Software Technology handlar i mångt och mycket om följande frågeställning:

> *Hur designar man programmeringsspråk för olika ändamål?*

Ändamålen för ett programmeringsspråk kan vara olika:

- Effektivitet
- Minska buggar
- Förhindra säkerhetsläckor
- Lättare att skriva större program
- Lättare att skriva en viss typ av program

Feldspar

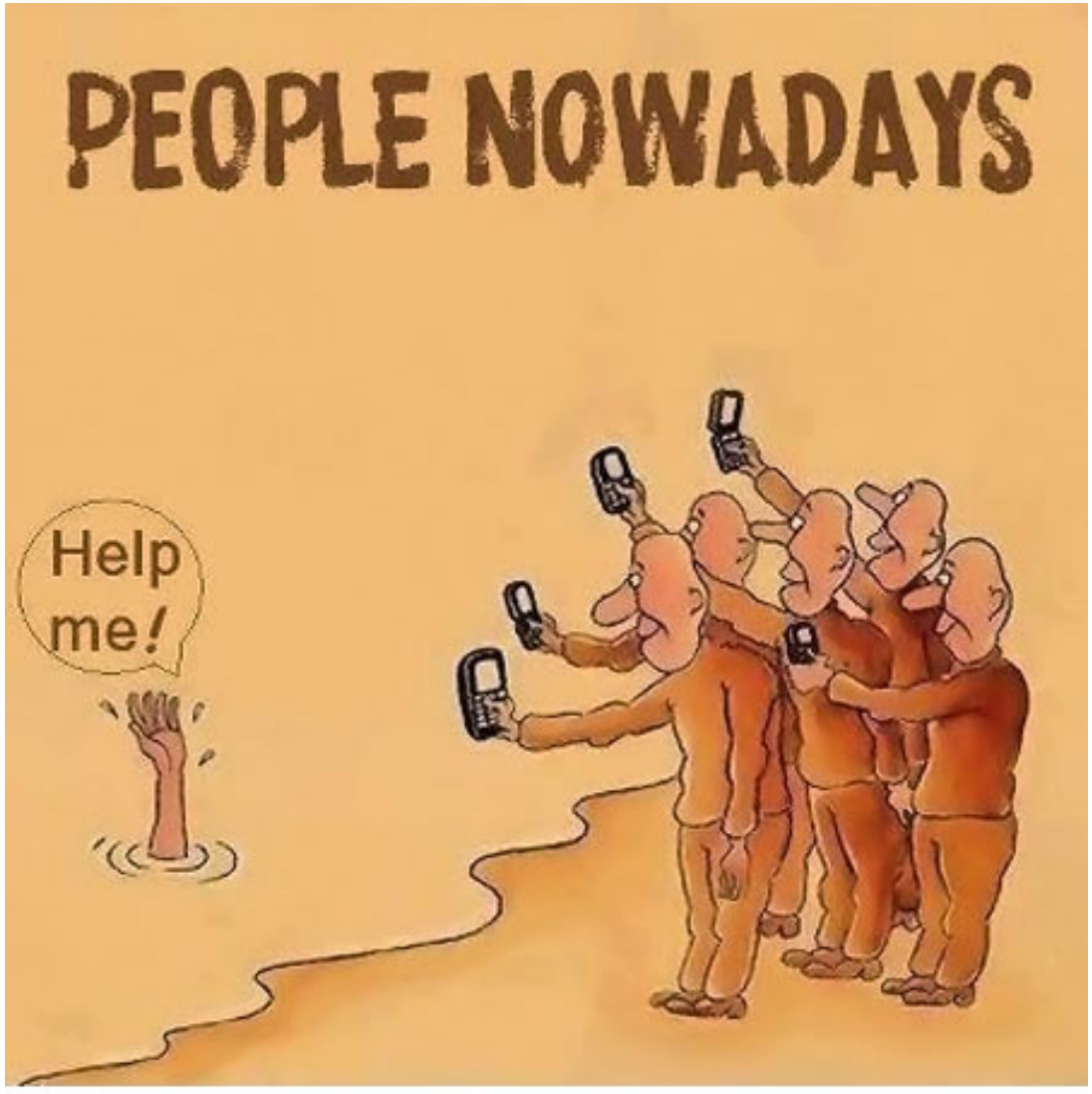
Domänspecifika språk

Ett domänspecifikt språk är ett språk som är inriktat på ett specifikt område, eller *domän*.

Exempel:

- Matlab, array-programmering
- Excel, kalkylark
- Regexp, reguljära uttryck
- UnrealScript, beskriva beteenden i 3D-spel

Figure


PEOPLE NOWADAYS
Help me!
more awesome pictures at THEMETAPICTURE.COM

Figure


WHAT'S THE POINT OF BEING AFRAID
OF THE ZOMBIE APOCALYPSE
WHEN YOU'RE ALREADY A ZOMBIE?
more awesome pictures at THEMETAPICTURE.COM

Figure

Programmera basstationer

Hur basstationer programmeras idag:

- Lågnivå kod
- C och assembler

Problem:

- Svårt att använda koden på ny hårdvara
- Svårt att parallelisera koden
- Svårt att skriva korrekt kod

Feldspar

Feldspar

- Ett domänspecifikt språk, utvecklat på Chalmers i samarbete med Ericsson

Feldspar är designat för att underlätta utvecklingen av inbäddade system:

- Höja abstraktionsnivån för programmering av inbäddade system
- Generera effektiv kod för inbäddade system
- Generera parallel kod

Skalärprodukt

Skalärprodukt

```
float sum = 0.0;
for (int i=0;i<length;i++) {
  sum += a[i] * b[i];
}
```

Skalärprodukt

```
float sum = 0.0;
for (int i=0;i<length;i++) {
  sum += a[i] * b[i];
}
```

- Lätt att göra fel med array indexering
- Svårt att parallelisera
- Svårt att se vilken algoritm som implementeras

Skalärprodukt i Feldspar

```
scalarProd2 :: DVector Float -> DVector Float
            -> Data Float
scalarProd2 a b = sum (zipWith (*) a b)
```

Skalärprodukt i Feldspar

```
scalarProd2 :: DVector Float -> DVector Float
            -> Data Float
scalarProd2 a b = sum (zipWith (*) a b)
```

- Närmare den matematiska definitionen
- Inga array indexeringar som kan gå fel
- Enkelt att parallelisera
- Effektivitet?

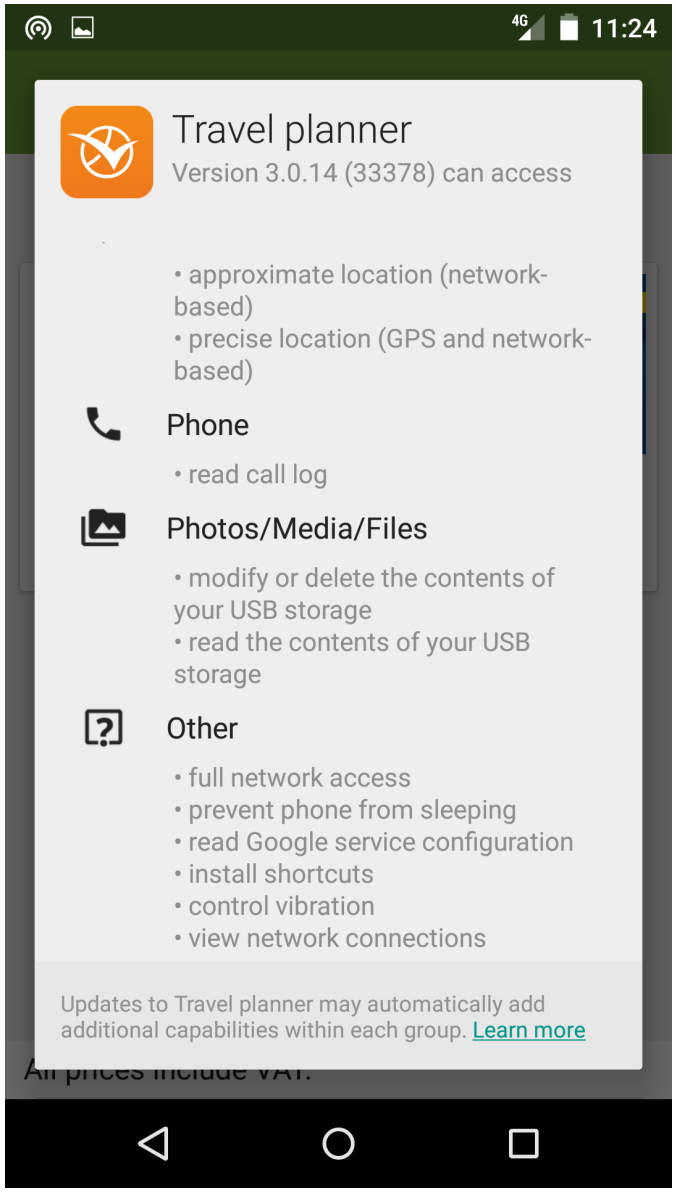
Fusion

- Feldspar har en optimering som kallas *fusion*.
- Den tar bort intermediära vektorer
- I fallet med skalärprodukten blir programmet en tight loop
- Gör att man kan skriva program på en hög nivå och ändå få effektiva program
- Feldspar är noga designat så fusion alltid kan optimera program

Fusion i praktiken

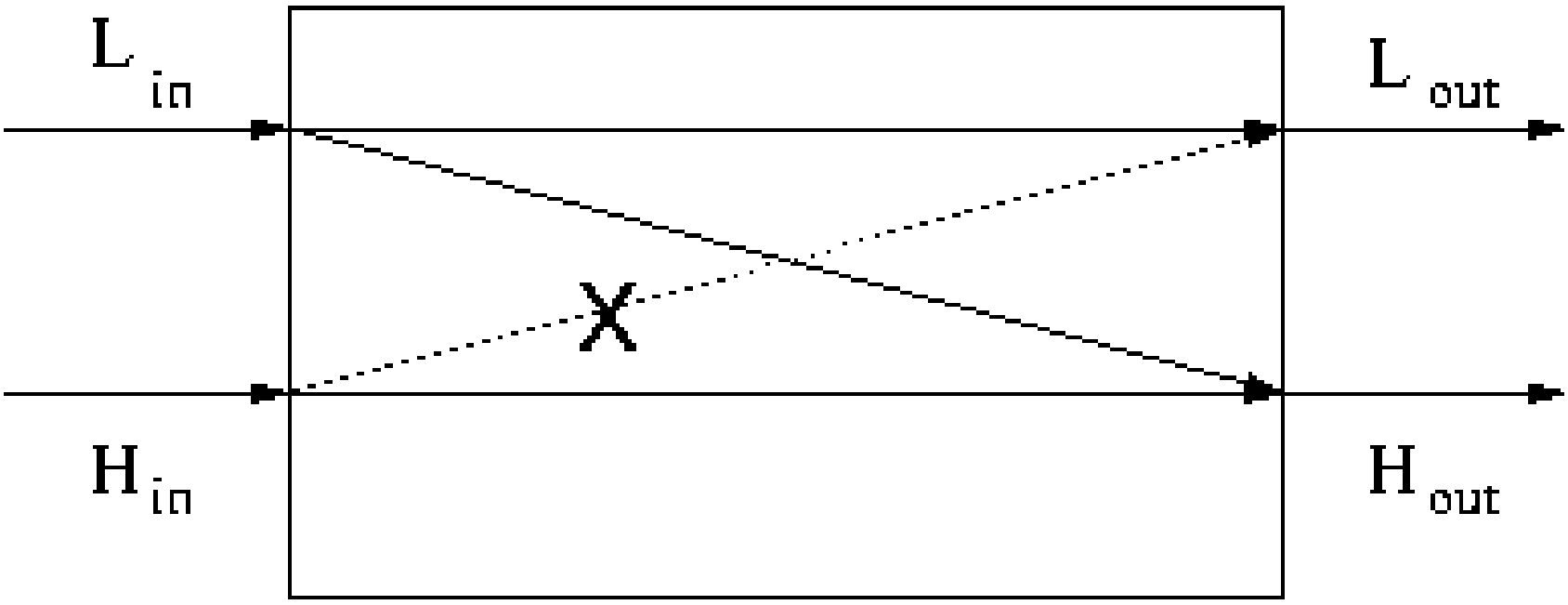
- Ericsson implementerade delar av 4G protokollet i Feldspar
- Feldspar-koden visade sig vara snabbare än Ericssons kod som används i produktion
- Fusion visade sig vara den avgörande faktorn

Säkerhet


11:24
Travel planner
Version 3.0.14 (33378) can access
• approximate location (network-based)
• precise location (GPS and network-based)
Phone
• read call log
Photos/Media/Files
• modify or delete the contents of your USB storage
• read the contents of your USB storage
Other
• full network access
• prevent phone from sleeping
• read Google service configuration
• install shortcuts
• control vibration
• view network connections
Updates to Travel planner may automatically add additional capabilities within each group. Learn more

Figure

Informationsflöde



Figure

Informationsflöde

```
int l, h;
l = h;
```

Informationsflöde

```
int l;
bool h;
if (h) {
  l := 3
} else {
  l := 42
}
```

Paragon

Paragon är en utökning av Java

- Hjälper att skriva säkra program som inte läcker information av misstag
- Kan uttrycka mycket kraftfulla säkerhetspolicies, t.ex. deklassificering

JSFlow

- En implementation av Javascript för att spåra informationsflöde
- Kan användas som en utökning i en browser

Race condition

```
class Race implements Runnable {

    int counter = 0;

    public void run() {
        for(int i=0; i<100000;i++) {
            counter++;
        }
    }
```

```
    public static void main(String[] args) {
        try {
            Race r = new Race();

            Thread t1 = new Thread(r);
            Thread t2 = new Thread(r);
            t1.start();
            t2.start();
            t1.join();
            t2.join();
            System.out.println("counter :" + r.counter);
        } catch (Exception e) {
            System.out.println(":-(");
        }
    }

}
```

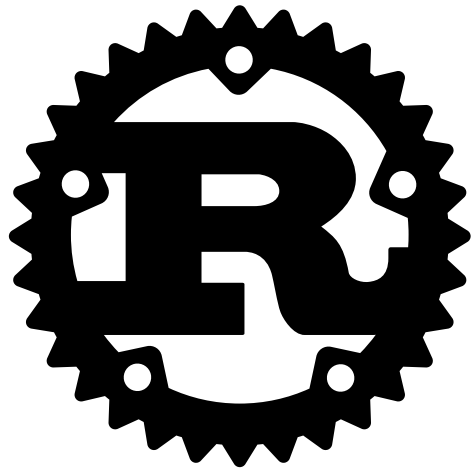
Race condition

Live demo

Thread1	Thread2	counter
		0
read		0
	read	0
increase		0
	increase	0
write		1
	write	1

mozilla

Rust



Figure

Rust

- Rust är ett nytt programmeringsspråk från Mozilla
- Vill kunna generera minst lika effektiv kod som C++
- Vill kunna skriva
- Vill kunna undvika många typer av programmeringsfel
  - Race conditions
  - Typfel
  - Minnesläckor
  - ...
- Servo, nästa generations webläsare
  Rust används för att skriva en ny webläsare från Mozilla

Exjobb

- Vi har för närvarande en exjobbare som jobbar med att förbättra stödet för parallelism i Rust.
- Projektet kommer att förbättra stödet för s.k. *task-parallelism* i Rust

Slutresultatet kommer att användas bl.a. för att snabba upp renderingen av websidor i Servo.

Kurser

Kompilatorer

- Programming Language Technology, DAT151
- Compiler Construction, TDA283

Ni får lära er:

- Hur en kompilator översätter ett språk till bytekod/maskinkod
- Hur ni skriver en egen kompilator
- En djupare förståelse för hur program exekverar på en maskin

Language Based Security

- Language Based Security, TDA602

Ni får lära er om:

- Olika typer av attacker, t.ex. buffer overrun, code injection
- Hur man programmerar för att undvika attacker
- Hur moderna programmeringsspråk kan hjälpa att skriva säkrare program

Programmeringsspråk

- Frontiers of Programming Languages, TDA261
- Programming Paradigms, DAT121

Ni får lära er om:

- Olika paradigmer för programmeringsspråk och vad de är bra för
- Vad som krävs för att designa ett programmeringsspråk
- Verktyg för att manipulera program

Funktionell Programmering

- Functional Programming, TDA452
- Advanced Functional Programming, TDA342

Ni får lära er om:

- Paradigmed Funktionell Programmering och dess fördelar
- Programmering i språket Haskell
- Använda Haskell för att skriva *inbäddade* domänspecifika språk

Många programmerssspråk, t.ex. C++, Java, C#, har på sistone valt att inkludera koncept från funktionell programmering

Parallel Programmering

- Parallel programmering (Concurrent Programming), TDA383

Ni får lära er om:

- Olika problem med parallel programmering, t.ex. race conditions
- Olika metoder för att hantera problemen med
  - Semaforer
  - Monitorer
  - Message passing
  - Software Transactional Memory

Parallel Funktionell Programmering

- Parallel Functional Programming, DAT280

Ni får lära er om:

- Hur man kan snabba upp kod genom att parallelisera den
- Olika paradigmer för parallel programmering, t.ex. message passing, map-reduce, data parallelism, gpu programmering, parallel skeletons
- Hur funktionell programmering kan förenkla parallel programmering