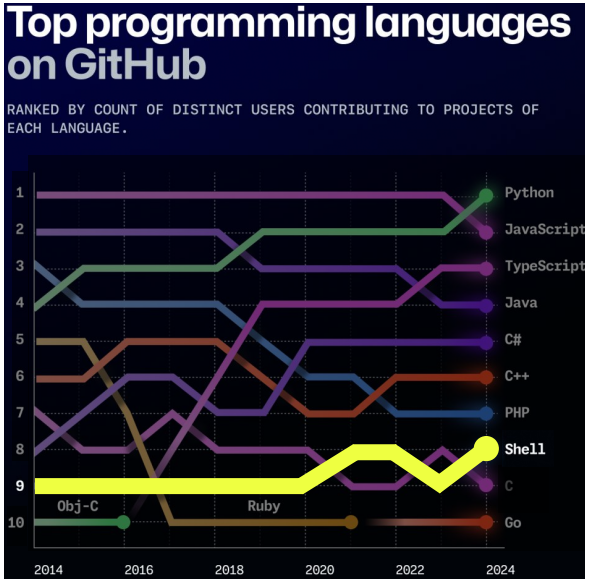
Also here today!

From Ahead-of- to Just-in-Time and Back Again: Static Analysis for Unix Shell Programs

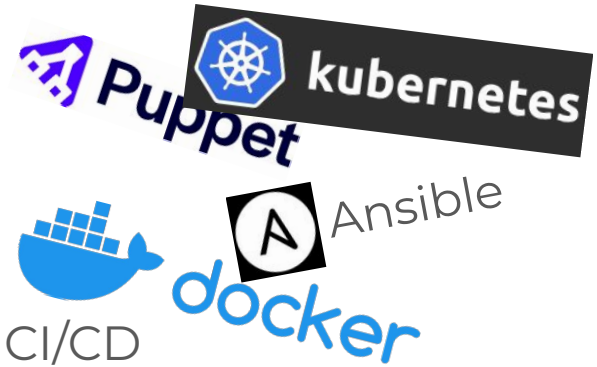
Lukas Lazarek, Seong-Heon Jung,
Evangelos Lamprou, Zekai Li,
Anirudh Narsipur, Eric Zhao,
Michael Greenberg, Konstantinos Kallas,
Konstantinos Mamouras, Nikos Vasilakis
lukas_lazarek@brown.edu


The shell is enduringly popular...



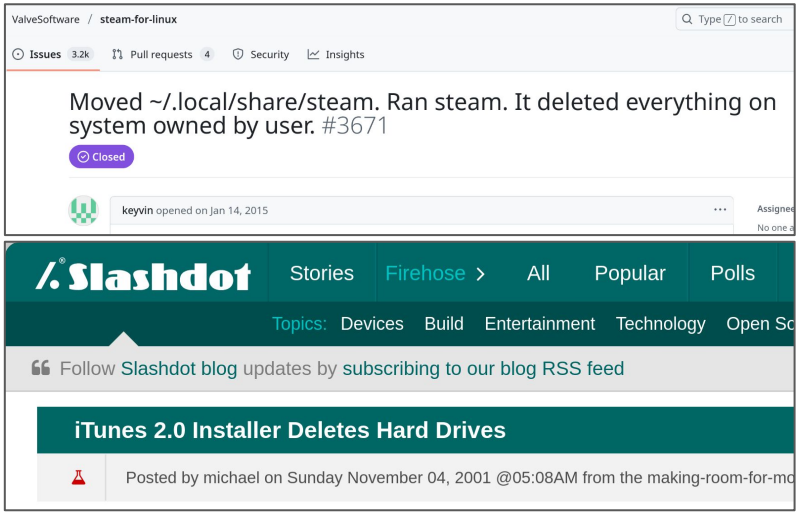
10

Many more



CI/CD

...and surprisingly complex!



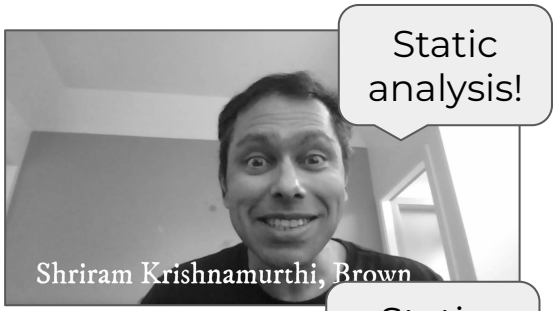
developers

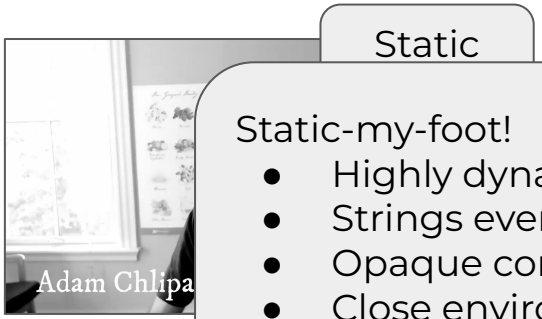
users

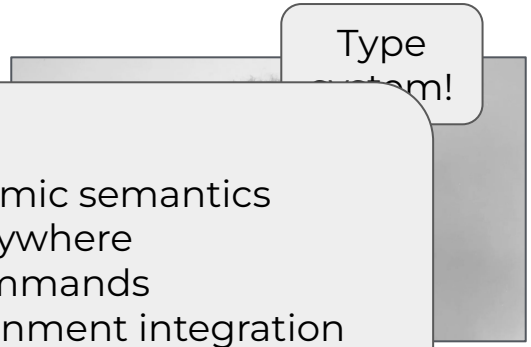
Can the shell have ahead-of-time support on par with other production languages?

For the benefit of both developers and users of Unix/Linux shell programs!

Flashback: Unix: the Next 50 Years [HotOS'21]







Static-my-foot!

- Highly dynamic semantics
- Strings everywhere
- Opaque commands
- Close environment integration

We **can** get ahead-of-time support on par with other production languages!

For the benefit of both developers and users of Unix/Linux shell programs!

Static analysis is feasible: key ideas

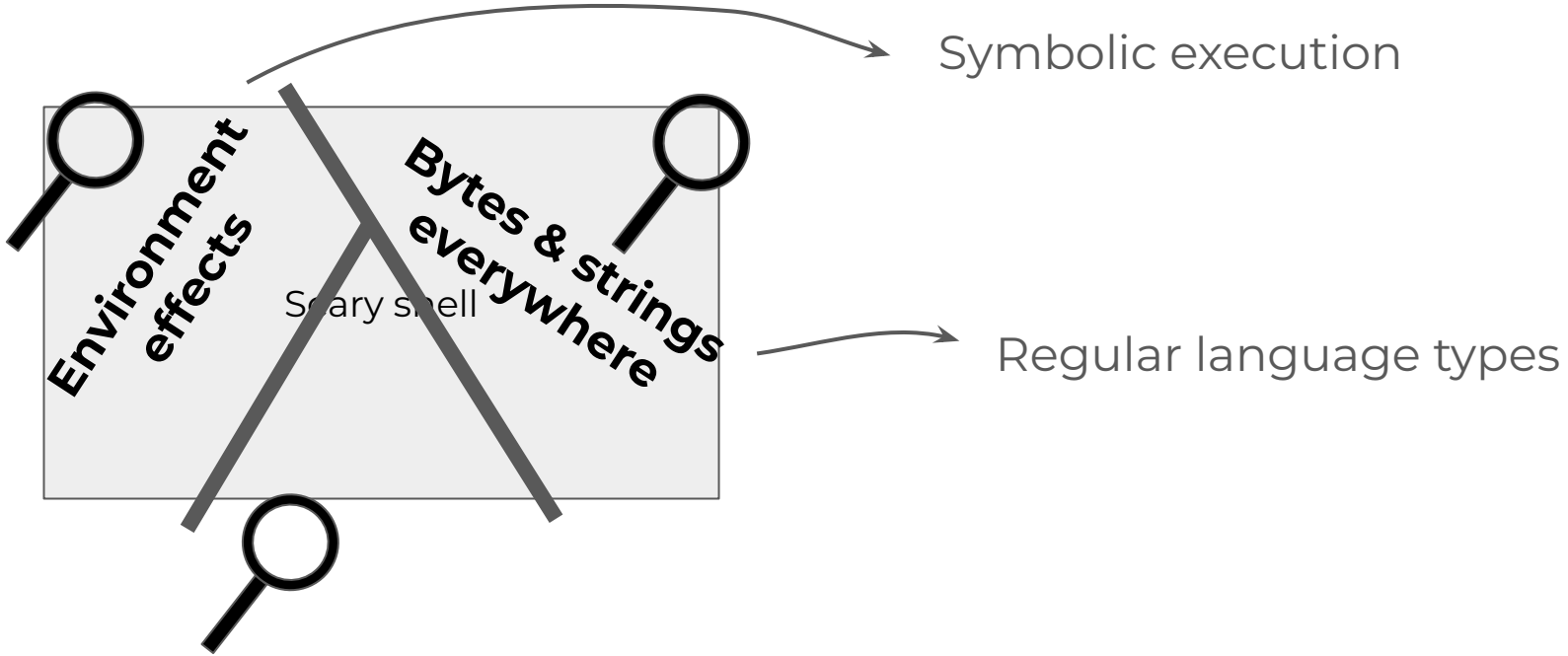
1. Divide & conquer: build static analysis subsystems for key subproblems

2. Trust, but verify: extract info from docs, then check it

3. Better late than sorry: apply checks at the right time, pre-catastrophe

1. Divide & conquer:
   specialize static analyses to subproblems

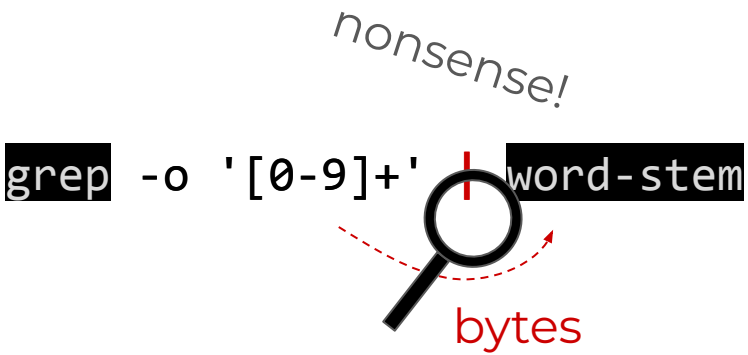
1) Divide & conquer: specialize static analyses to subproblems



Subproblem 1: byte communication

```
grep -o '[0-9]+' | word-stem
```

Subproblem 1: byte communication



Subsystem 1: regular types

grep :: .* → [0-9]+

```
grep -o '[0-9]+' | word-stem
```

Subsystem 1: regular types

① ②

grep :: .* → [0-9]+

```
grep -o '[0-9]+' | word-stem
```

① Input: .*
lines with 0 or more occurrences of anything

② Output: [0-9]+
lines with 1 or more occurrences of any of the characters 0-9

Subsystem 1: regular types

grep :: .* → [0-9]+ word-stem :: [A-Za-z]+ → [A-Za-z]+

```
grep -o '[0-9]+' | word-stem
```

Subsystem 1: regular types

grep :: .* → [0-9]+ word-stem :: [A-Za-z]+ → [A-Za-z]+

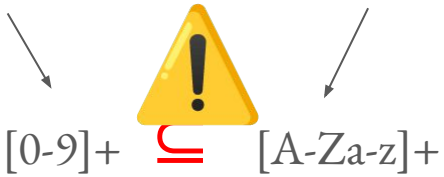
```
grep -o '[0-9]+' | word-stem
```

Composition problem!
Incompatible input for `word-stem`

Subsystem 1: regular types

grep :: $.^* \to [0\text{-}9]+$ word-stem :: $[A\text{-}Za\text{-}z]+ \to [A\text{-}Za\text{-}z]+$

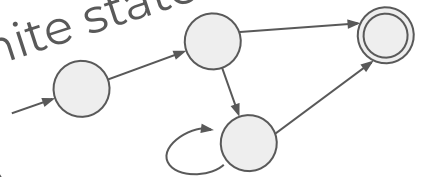
```
grep -o '[0-9]+' | word-stem
```



Composition problem!
Incompatible input for `word-stem`

More expressive types:

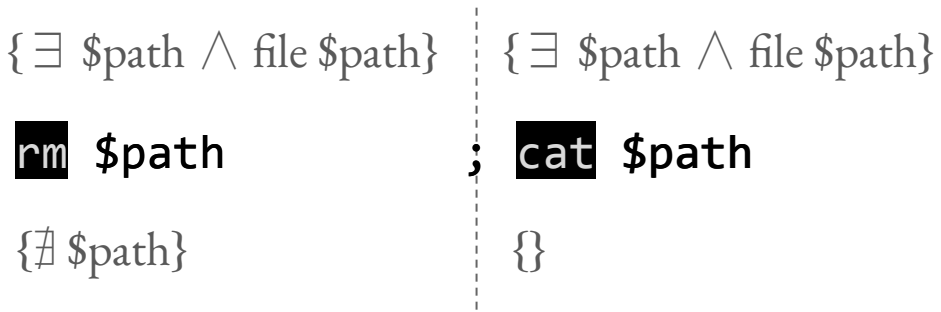
Finite state transducers



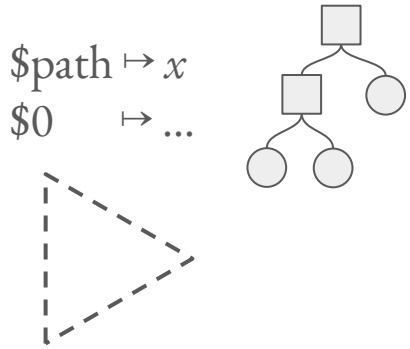
Polymorphism

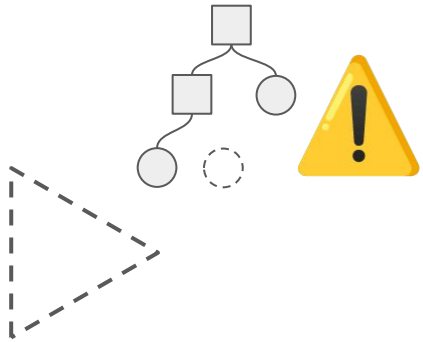
$\forall \alpha \subseteq [a\text{-}z]^* \cdot \alpha \to [0\text{-}9]+$

Subsystem 2: symbolic execution



Subsystem 2: symbolic execution





```
#!/bin/sh
if ...
    if ...
        if ...
            rm $path

# ... 100s of lines

cat $path
```

{∃ $path ∧ file $path}

```
rm $path
```

{♯ $path}

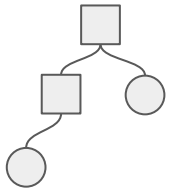
;

{∃ $path ∧ file $path}

```
cat $path
```

{}

```
path2=$(echo "${path%/*}" | cut -d/ -f3)
cat $path2
```

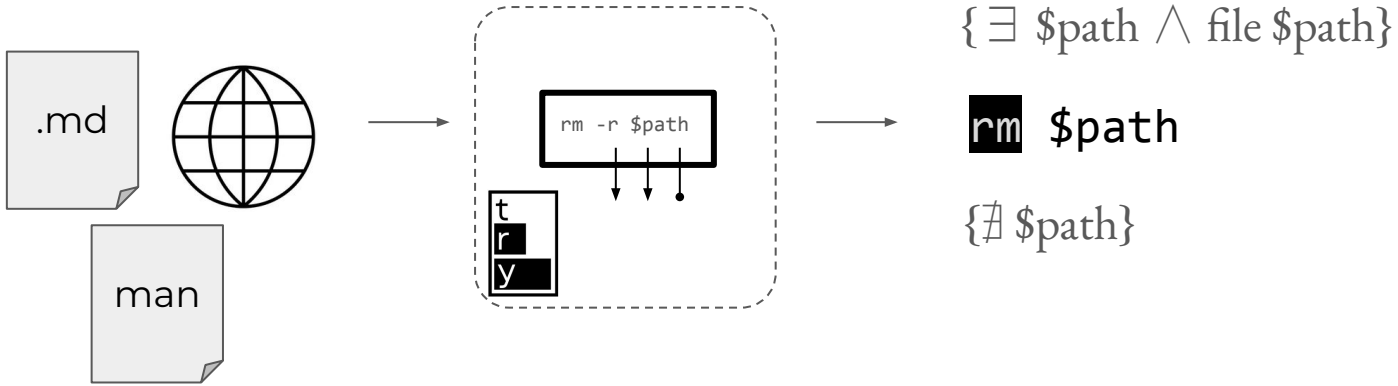


filesystem objects ⟷ path-string manipulation

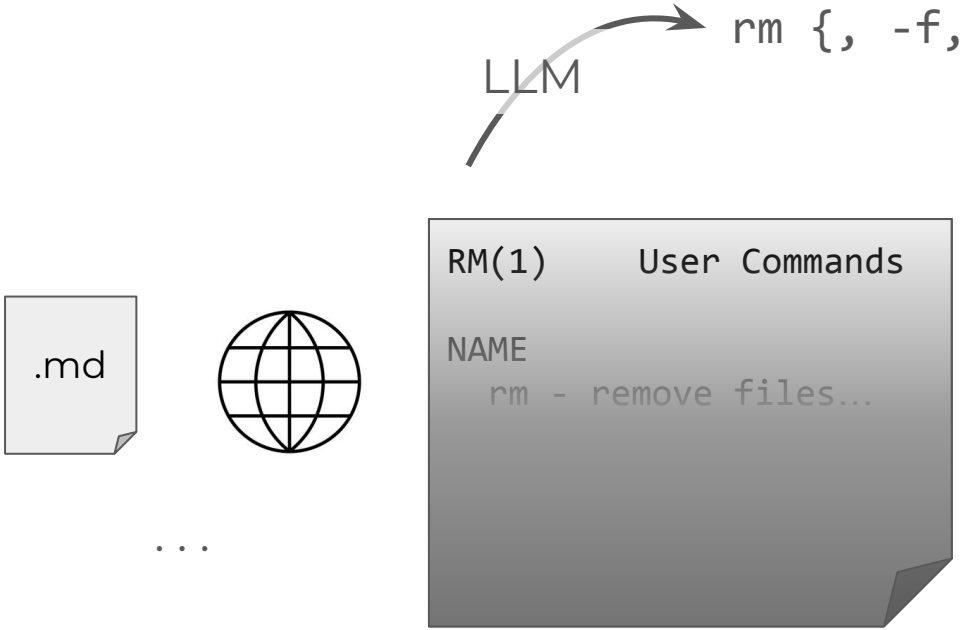
Automatic! SMT

2. Trust, but verify:
    extract information from docs, then check it

Trust, but verify: extract information from docs, then check it



Trust, but verify: extract information from docs, then check it



Static analysis is within reach

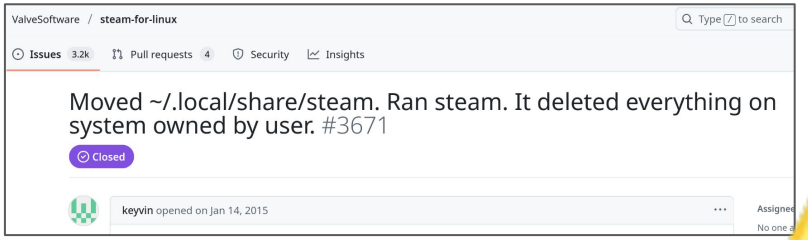
1. Divide & conquer: specialize static analyses to subproblems

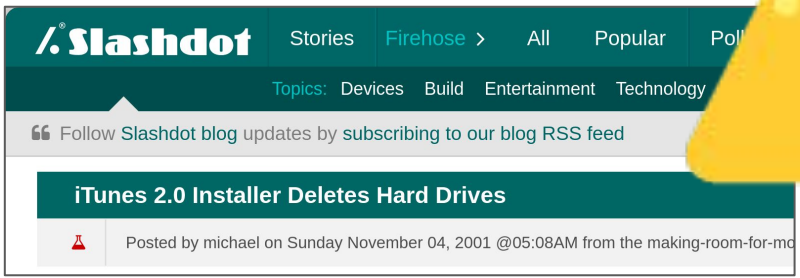
2. Trust, but verify: extract information from docs, then check it

3. Better late than sorry: apply checks at the right time, pre-catastrophe

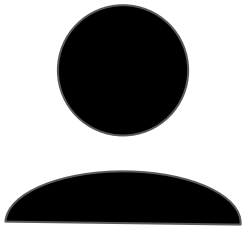
And it can prevent real catastrophes!

Prototype systems catch these bugs

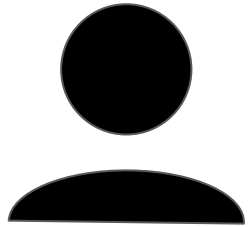




More sophisticated systems on the horizon...

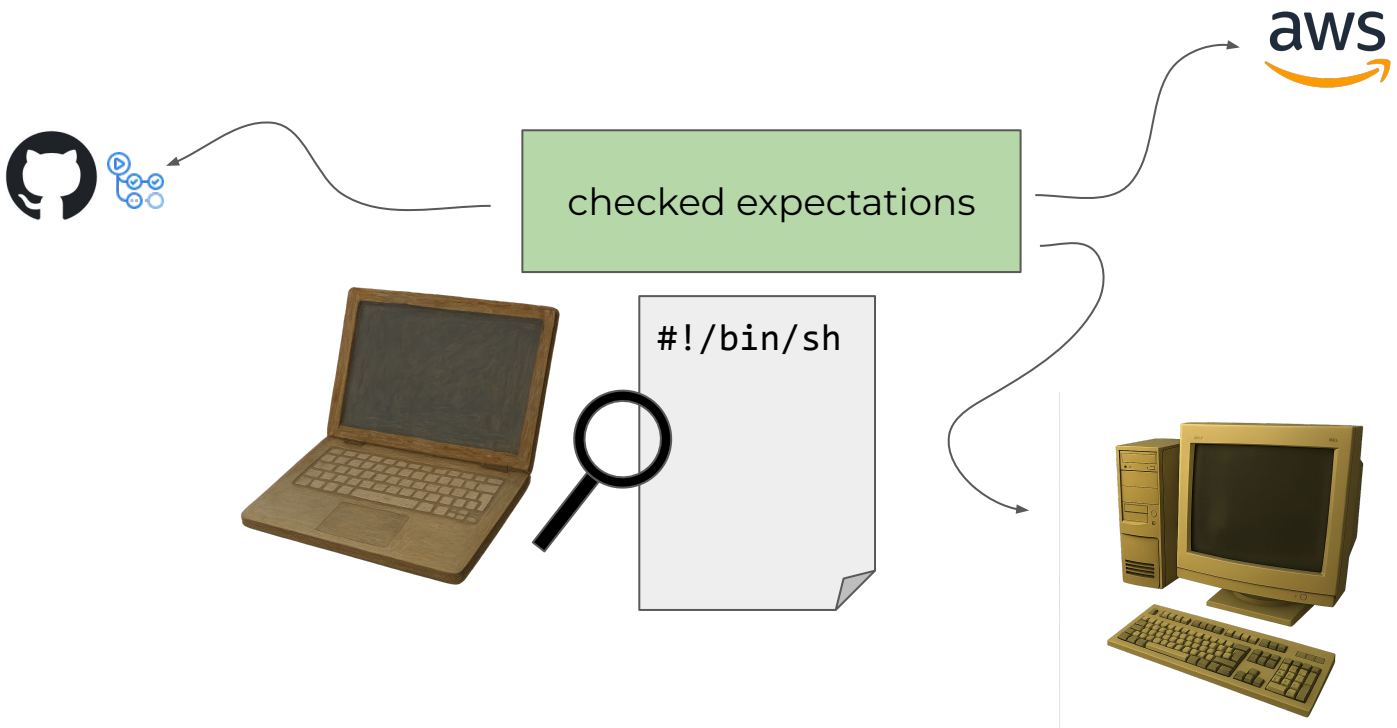


Developers

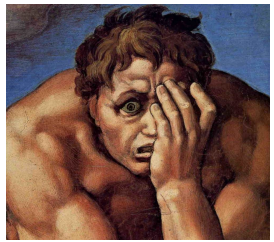


Users

Correctness: automatic script improvement



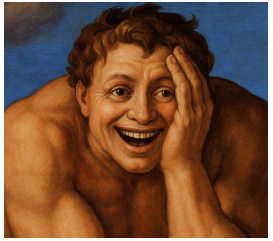
Checking and hardening



Software installation instructions:

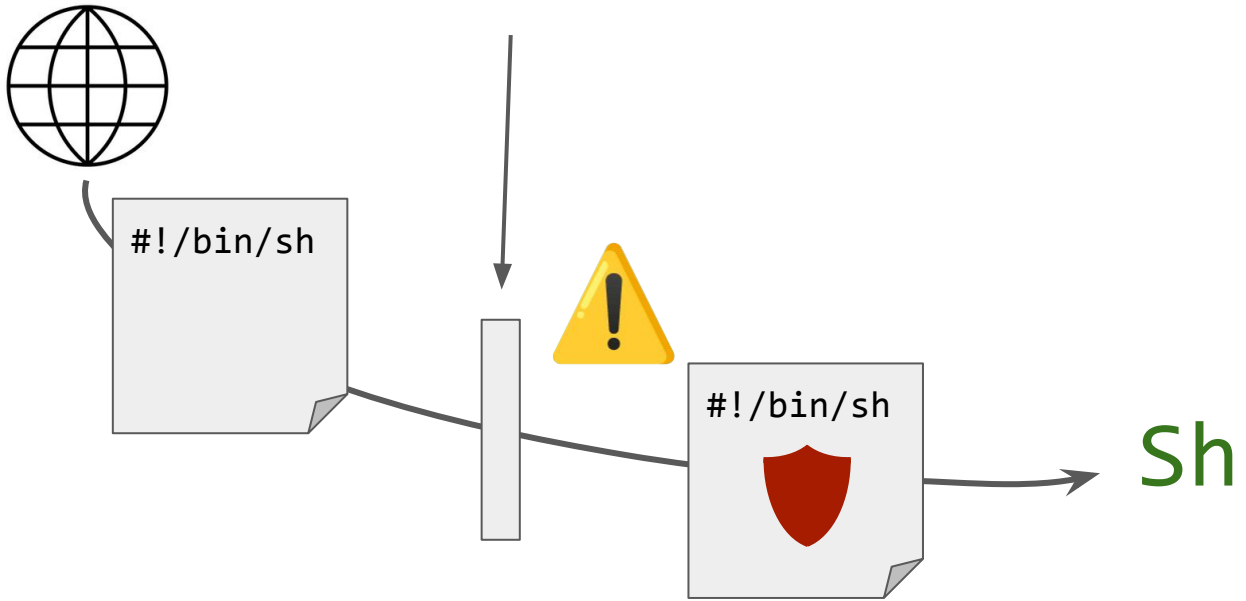
```
curl -sSf https://sh.rustup.rs | sh
```

Checking and hardening



Software installation instructions:

```
curl -sSf https://sh.rustup.rs | verify --noRW ~/personal | sh
```

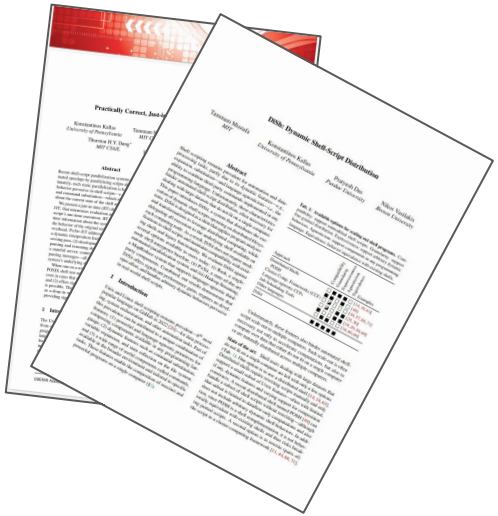


Performance

```
#!/bin/sh

for i in $(seq 1000); do
    curl .../data
    ./calc $i data > $i.out
done
```

Pash-JIT
DiSh
...



```
grep 'foo' file.txt | grep -v 'baz'

grep -P '^(?=.*foo)(?!.*baz)' file.txt
```

Static analysis for the shell is within reach

We **can** get ahead-of-time support on par with other production languages!

For the benefit of both developers and users of Unix/Linux shell programs!

1) Divide & conquer
2) Trust, but verify
3) Better late than sorry

Correctness

Security

Performance

Lukas Lazarek

ATLAS @ Brown

Get in touch!