

Spork

A posix_spawn you can use as a fork



Manuel Vögele



Christopher Thomas

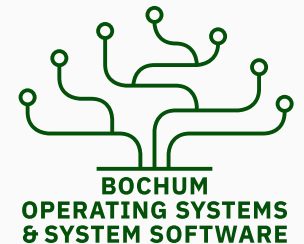


Timo Hönig

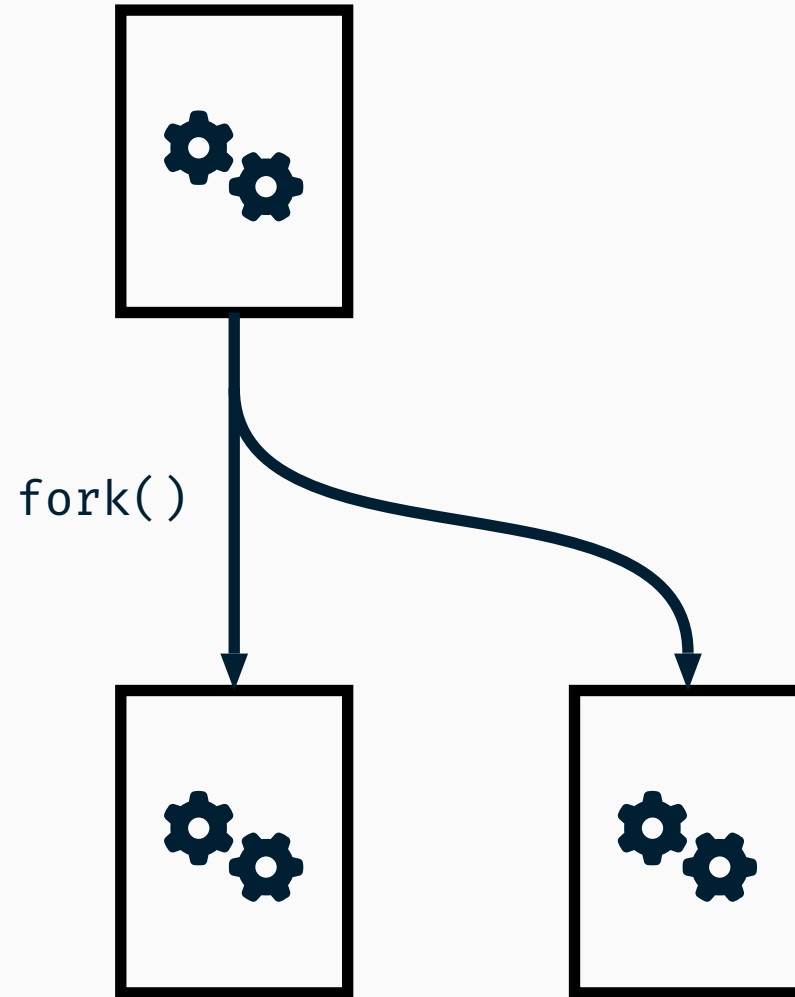
HotOS '25, 2025-05-15

RUHR
UNIVERSITÄT
BOCHUM

RUB



This work was partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – project number 539710462 (“DOSS”), 502228341 (“Memento”) and 465958100 (“NEON”) and from the Bundesministerium für Bildung und Forschung (BMBF, Federal Ministry of Education and Research) in Germany for the project SUSTAINET-inNOvAte 16KIS2262.





1971

1977

1983

1989

1995

2001

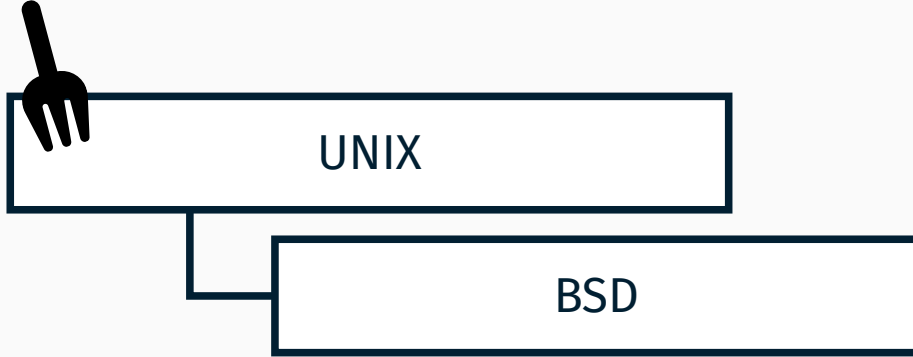
2007

2013

2019

2025

The History of Fork



1971

1977

1983

1989

1995

2001

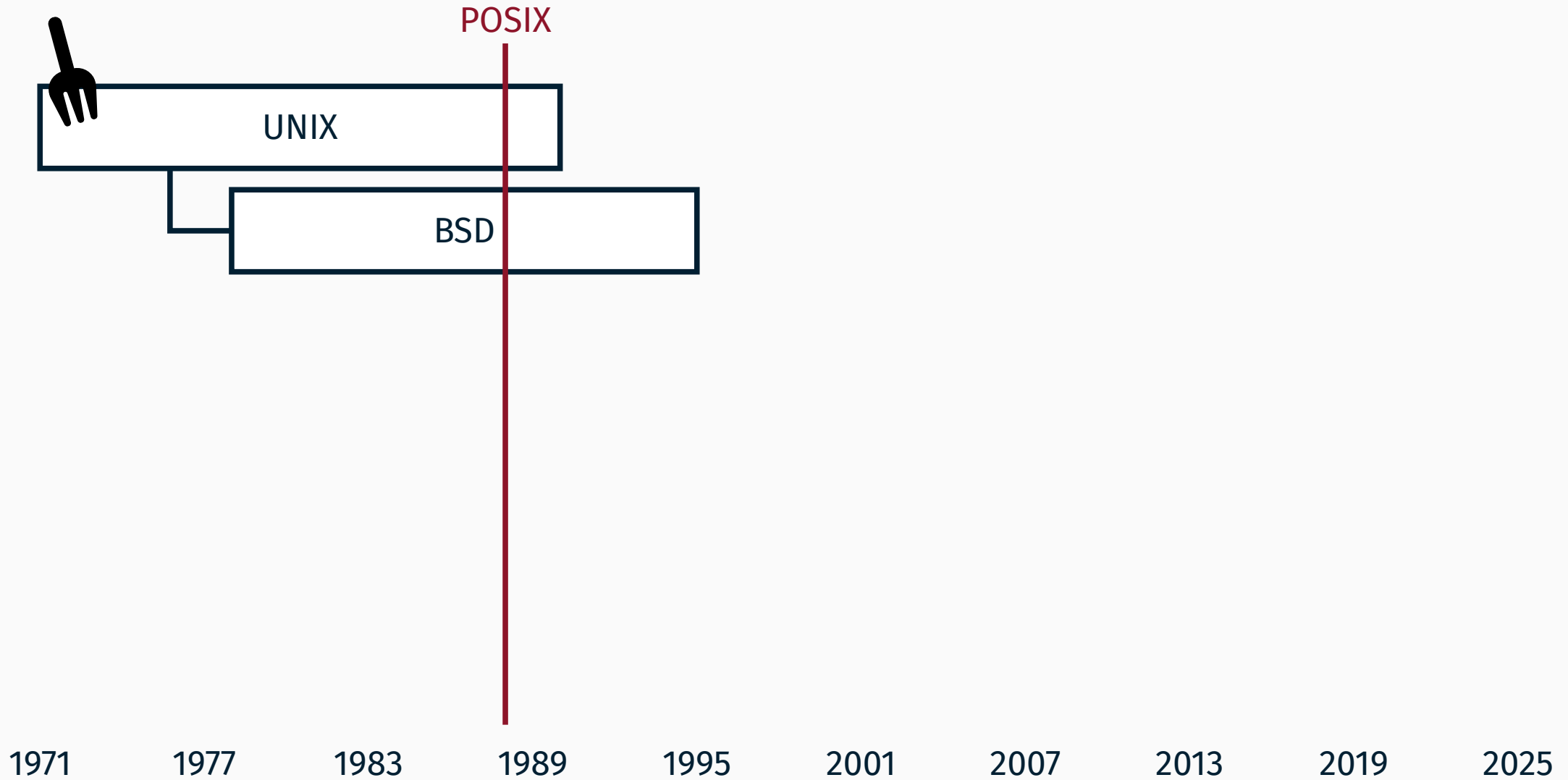
2007

2013

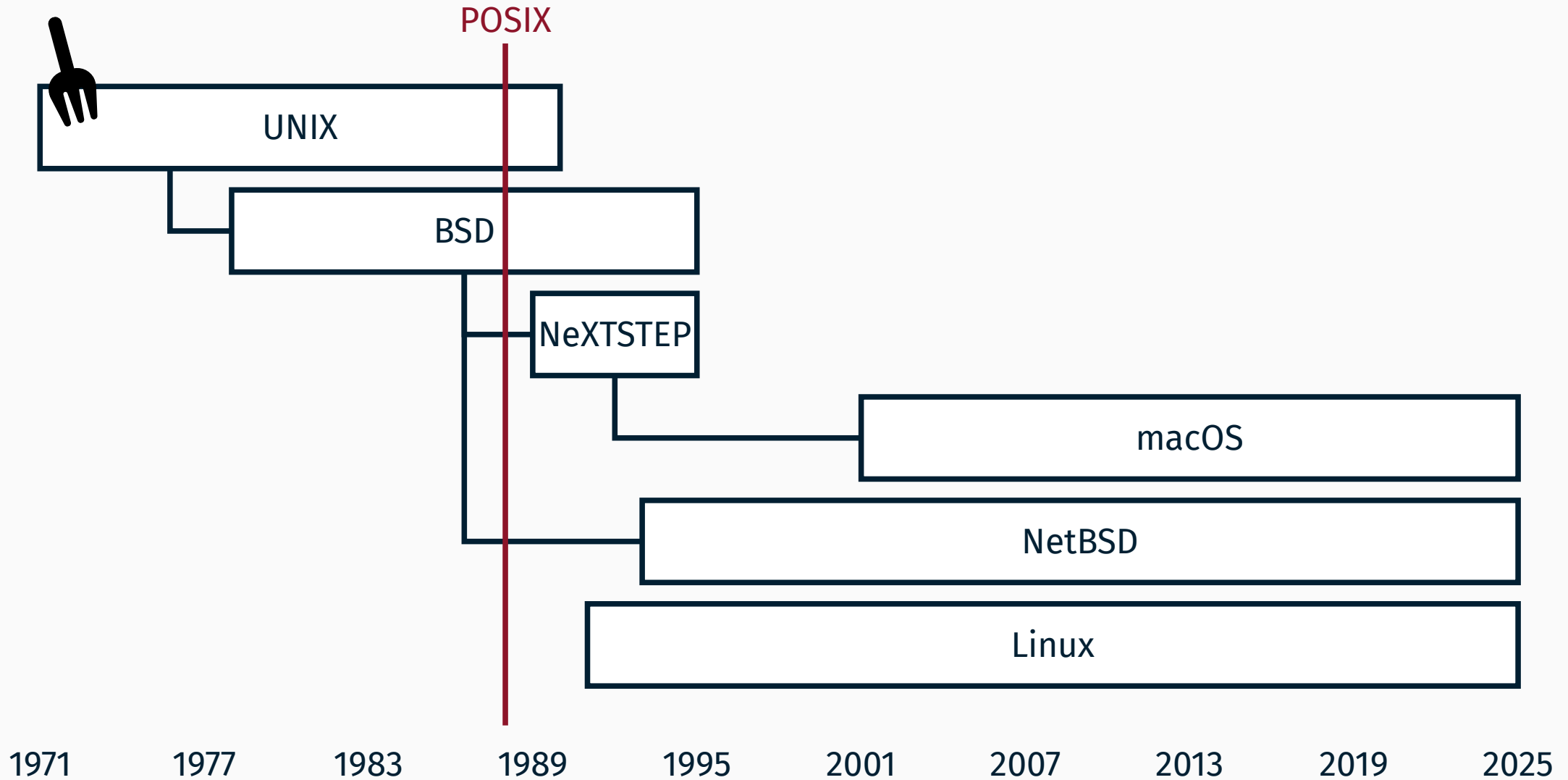
2019

2025

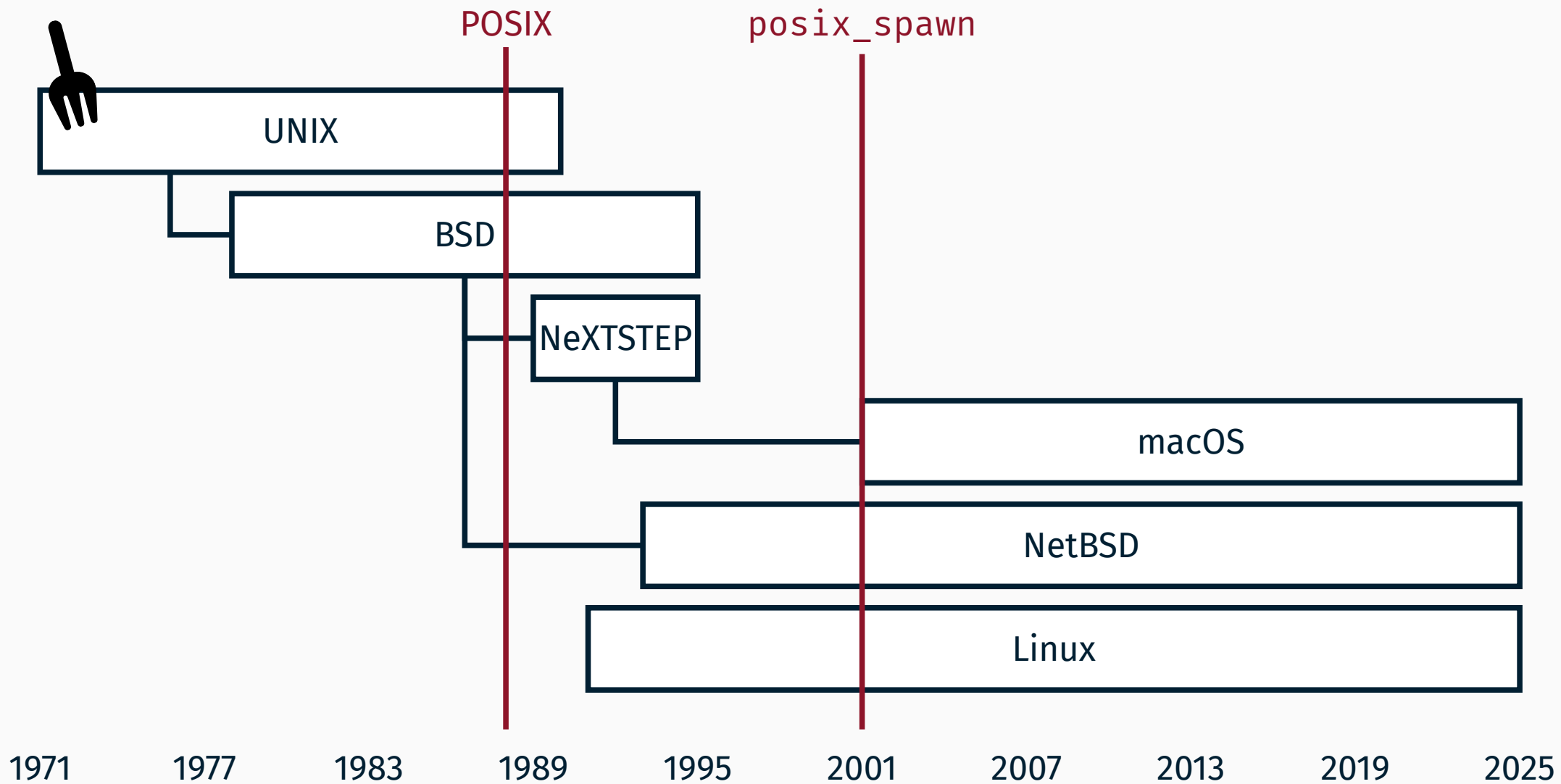
The History of Fork

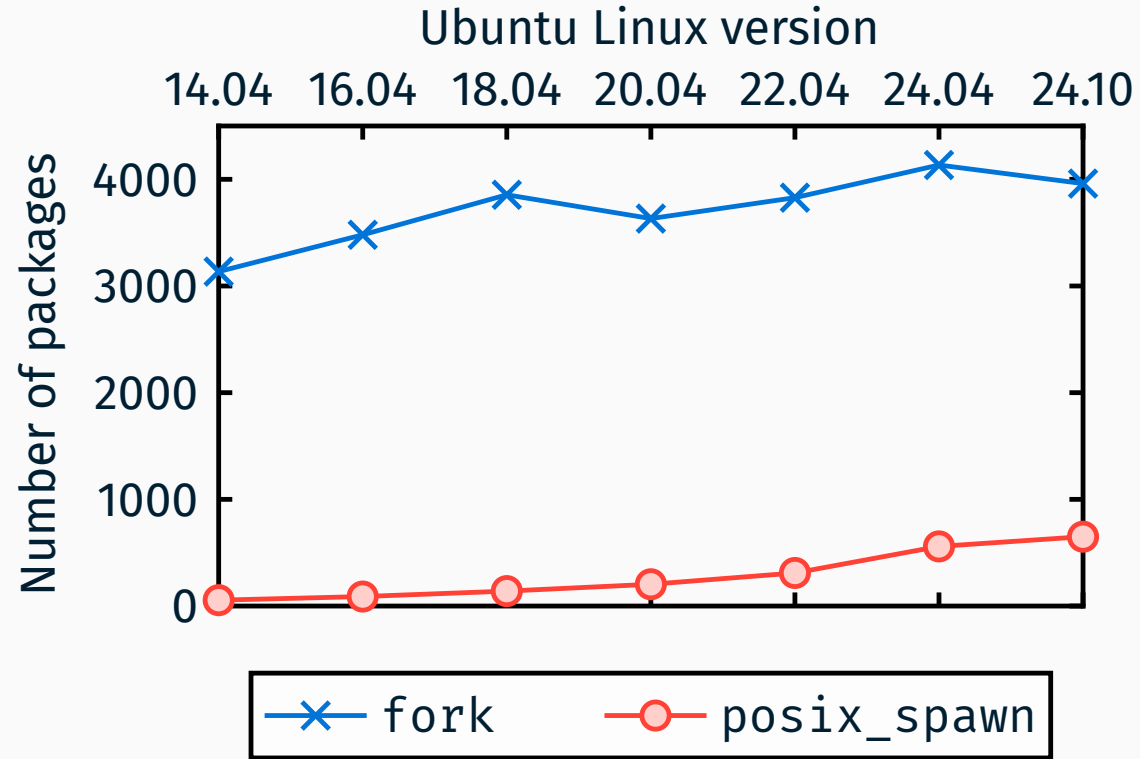


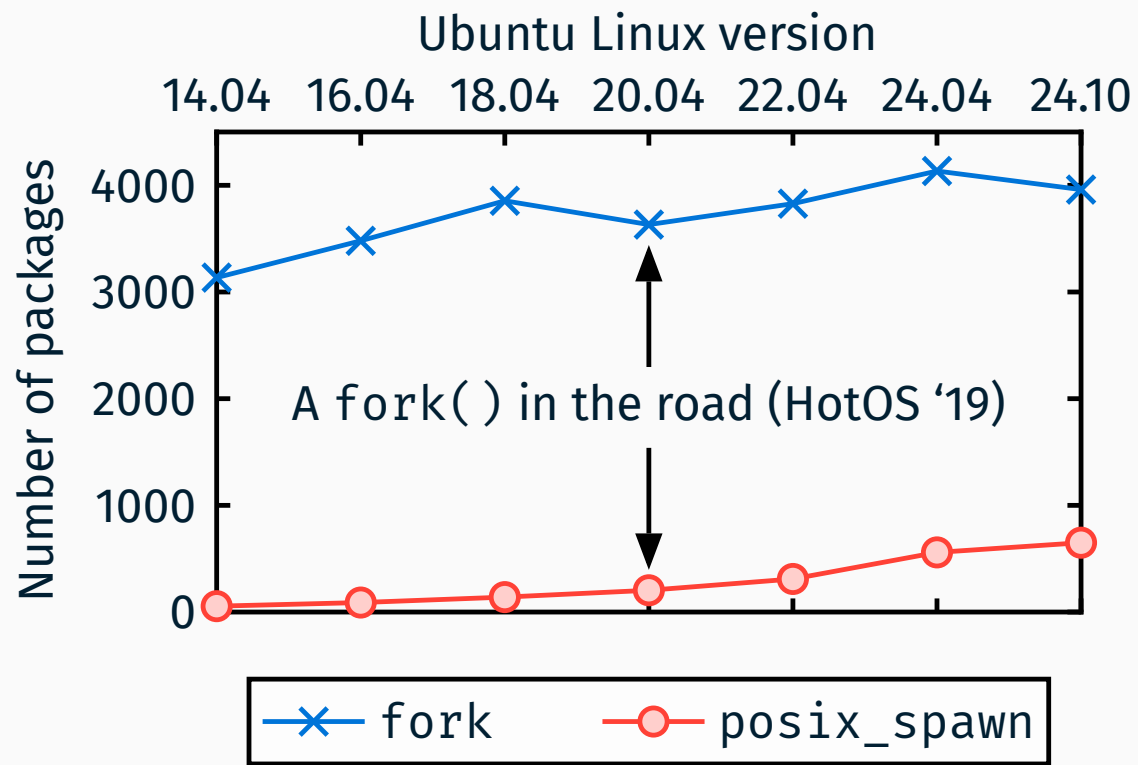
The History of Fork



The History of Fork









	fork
simple to implement	×
composes	×
thread-safe	×
secure API design	×
fast	×
scalable	×
no memory overcommit	×

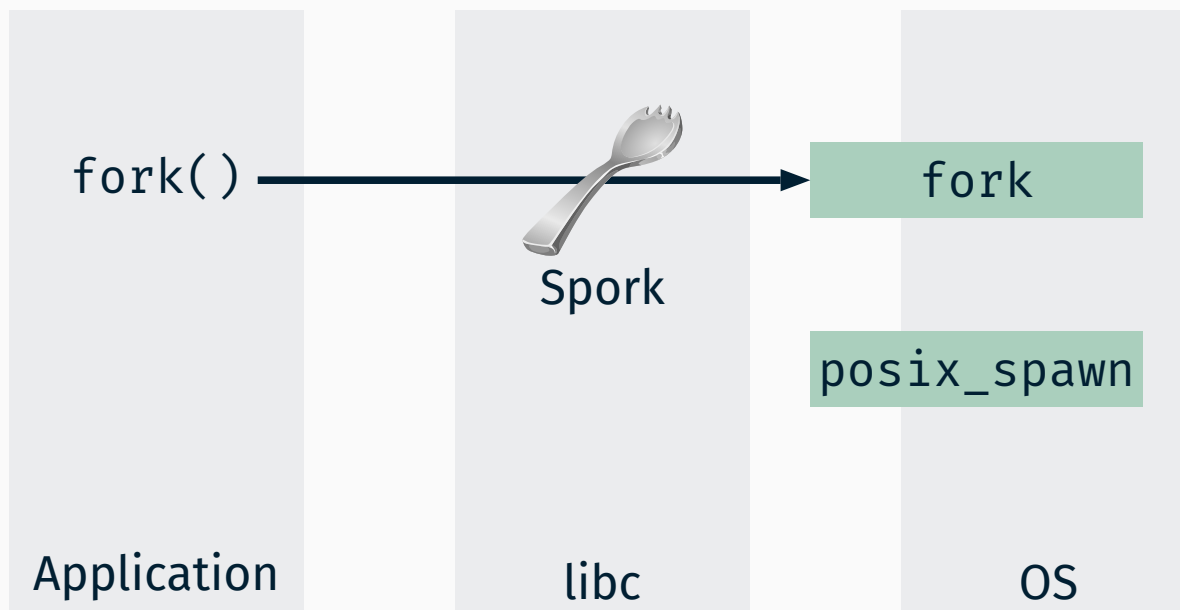


- `fork` is a bad syscall
- Applications need to be rewritten
- `posix_spawn` lacks functionality



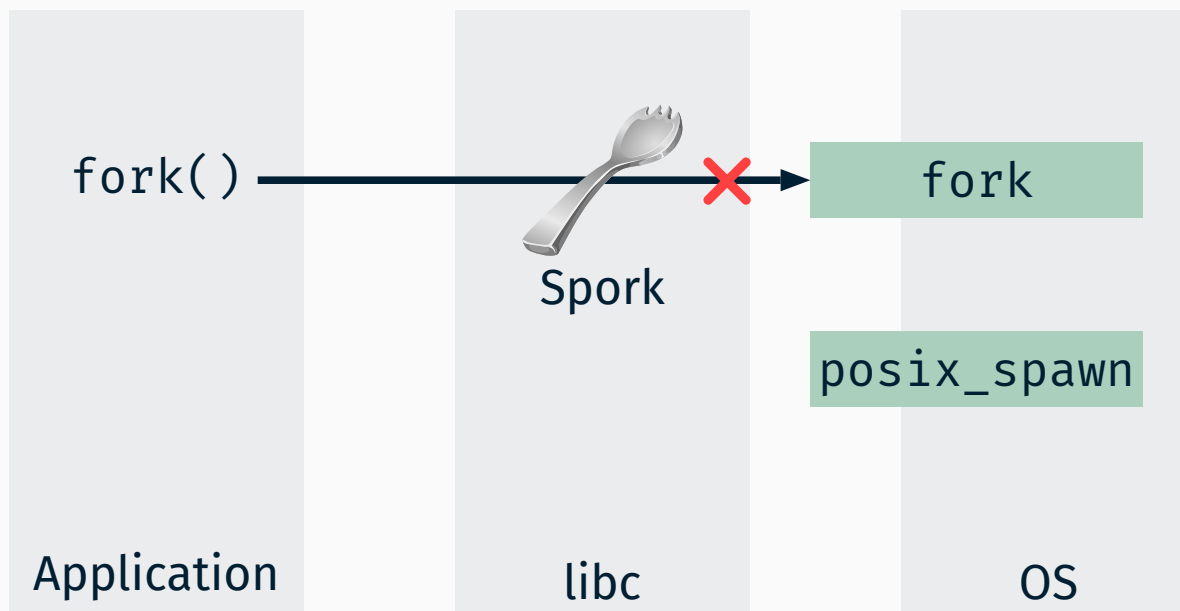


- `fork` is a bad syscall
- Applications need to be rewritten
- `posix_spawn` lacks functionality

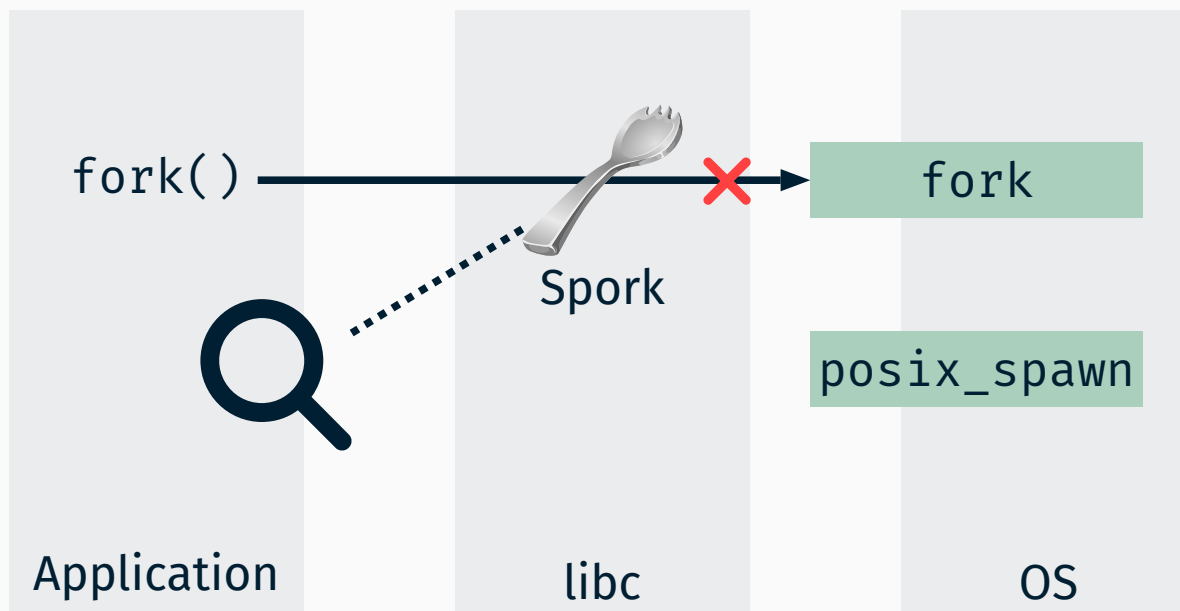




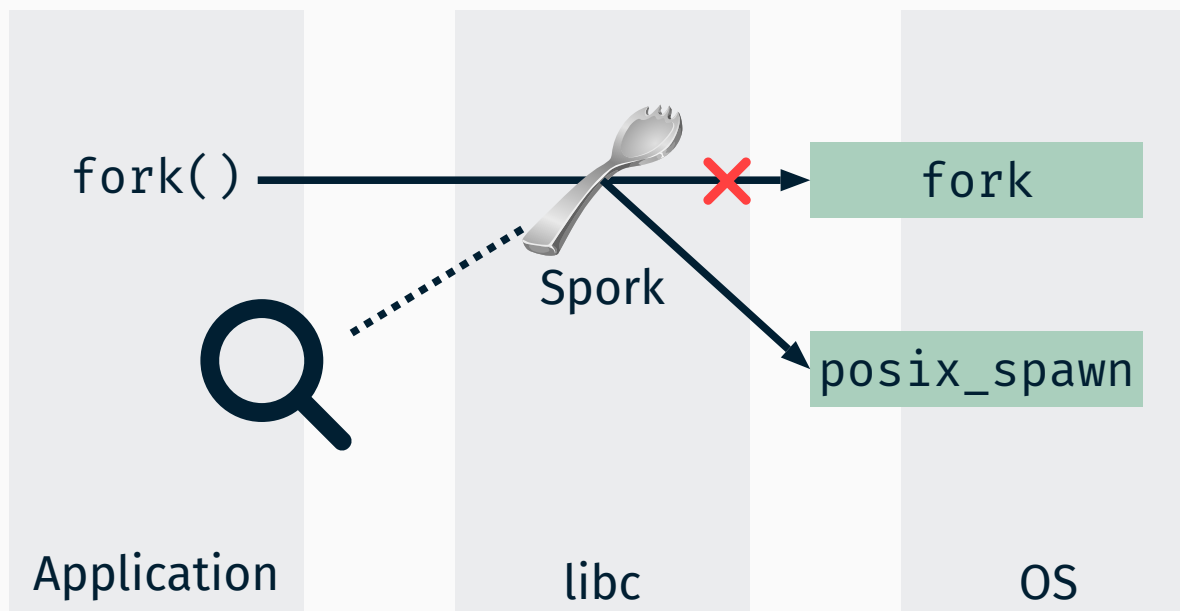
- `fork` is a bad syscall
- Applications need to be rewritten
- `posix_spawn` lacks functionality



- `fork` is a bad syscall
- Applications need to be rewritten
- `posix_spawn` lacks functionality



- `fork` is a bad syscall
- Applications need to be rewritten
- `posix_spawn` lacks functionality





Launch applications

- fork leads to an exec
- Process duplication mostly unnecessary



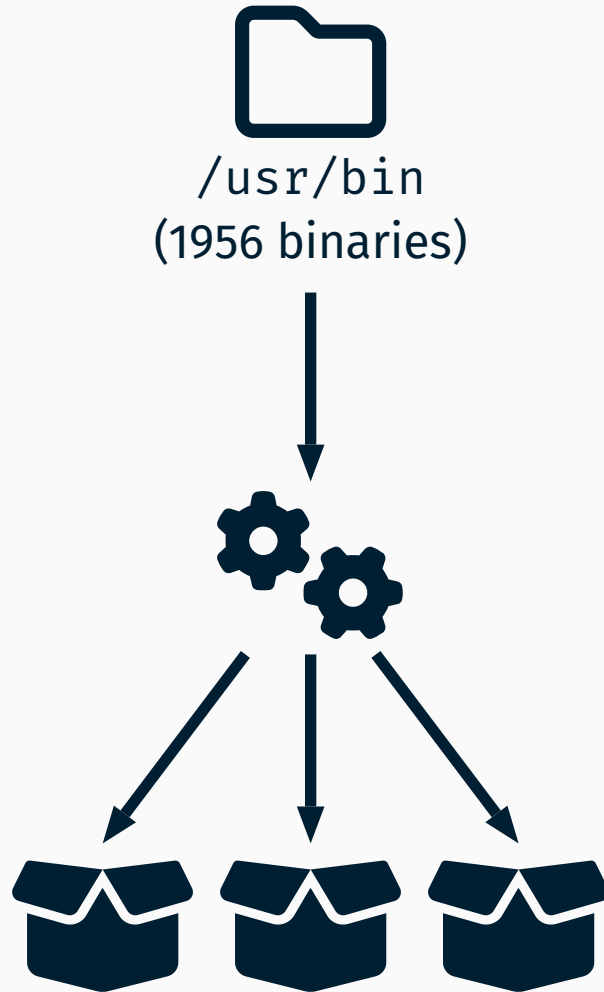
Create workers

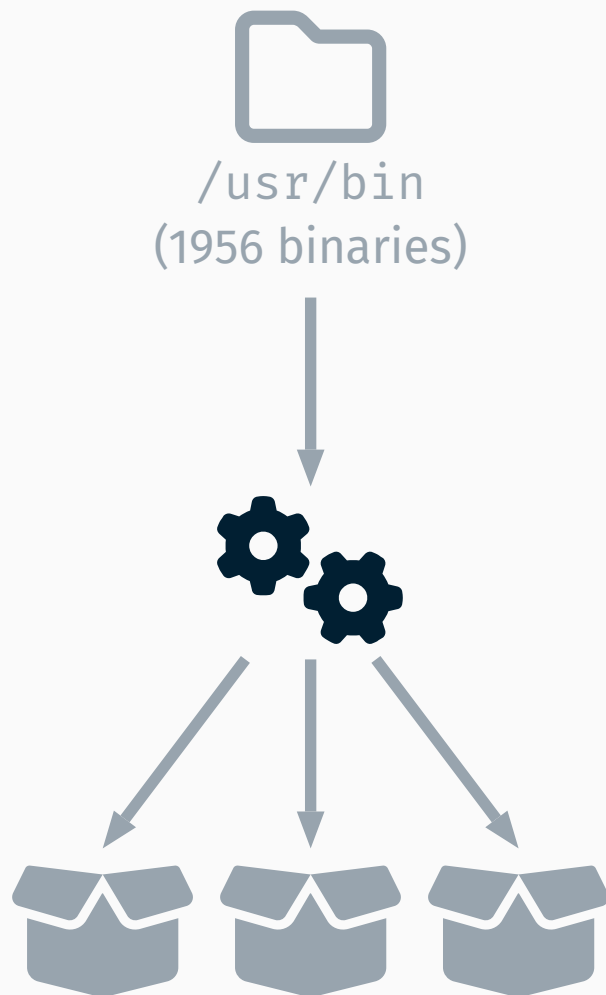
- Process duplication required
- Forked processes run in parallel
- Workers are created once and used for a long time



Create snapshots

- Process duplication required
- Forked processes remain dormant for extended periods of time
- Snapshots are created frequently
- Often relies on non-standard copy-on-write for performance





```
analyzed_code:
```

```
; ...
```

```
call fork
```

```
loop_head:
```

```
cmp rax, rcx
```

```
je prepare_exec
```

```
inc rcx
```

```
jmp loop_head
```

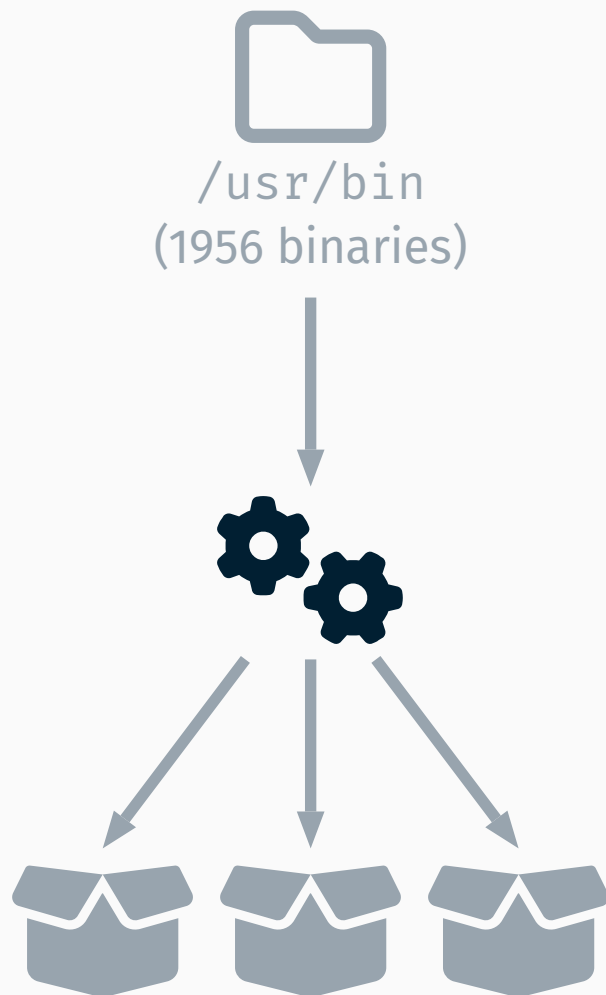
```
prepare_exec:
```

```
mov rdi, 0x1337
```

```
lea rsi, [rbp-0x20]
```

```
call execvp
```

```
ret
```



analyzed_code:

; ...

call fork

loop_head:

cmp rax, rcx

je prepare_exec

inc rcx

jmp loop_head

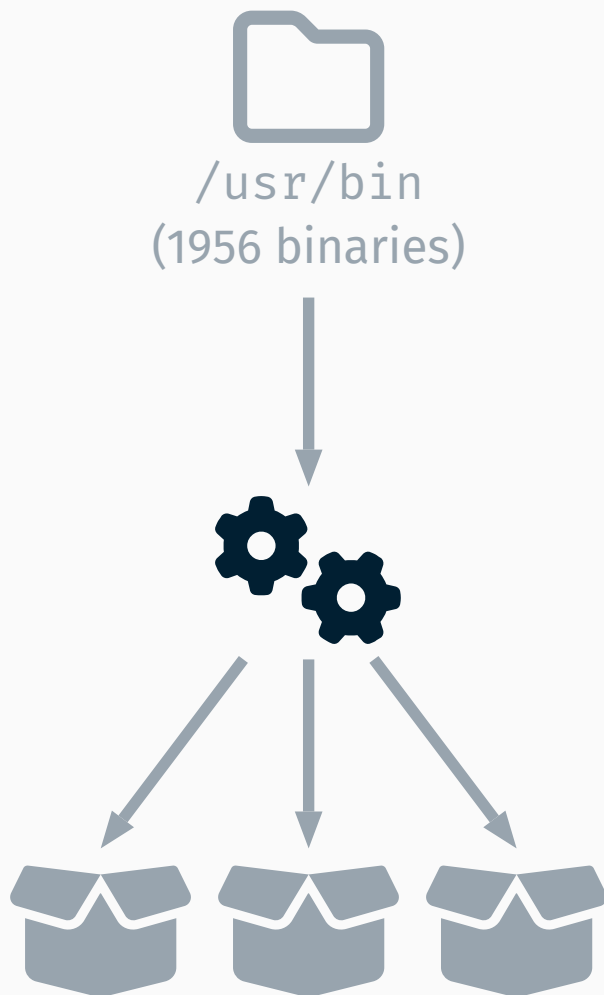
prepare_exec:

mov rdi, 0x1337

lea rsi, [rbp-0x20]

call execvp

ret



analyzed_code:

; ...

call fork

loop_head:

cmp rax, rcx

je prepare_exec

inc rcx

jmp loop_head

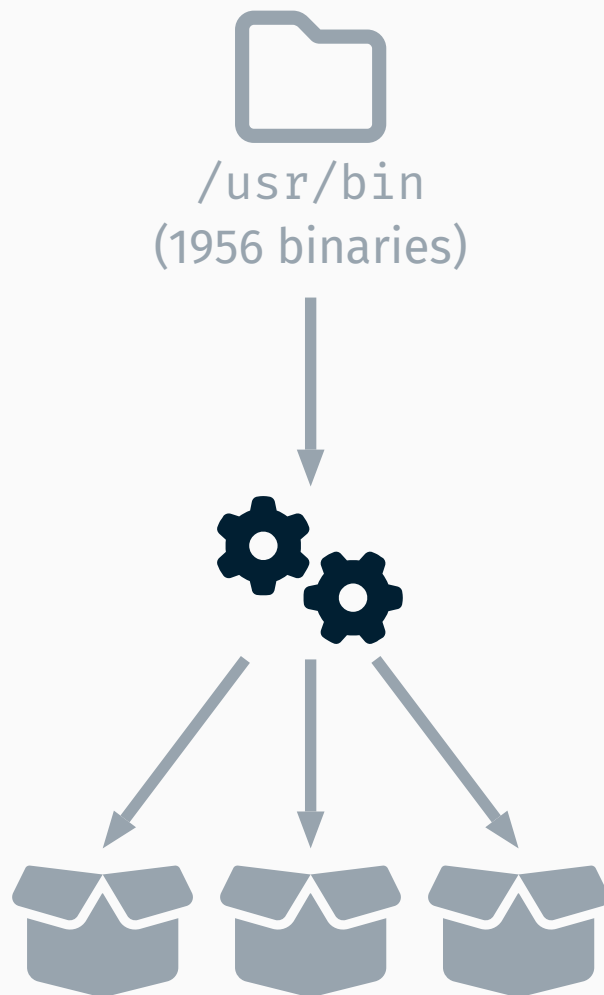
prepare_exec:

mov rdi, 0x1337

lea rsi, [rbp-0x20]

call execvp

ret



analyzed_code:

; ...

call fork

loop_head:

cmp rax, rcx

je prepare_exec

inc rcx

jmp loop_head

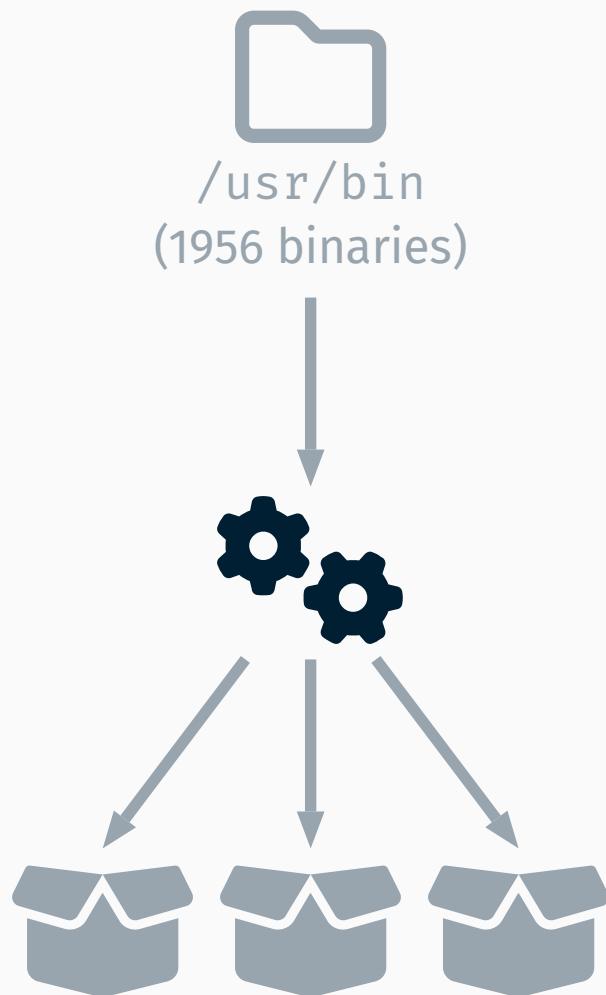
prepare_exec:

mov rdi, 0x1337

lea rsi, [rbp-0x20]

call execvp

ret



analyzed_code:

; ...

call fork

loop_head:

cmp rax, rcx

je prepare_exec

inc rcx

jmp loop_head

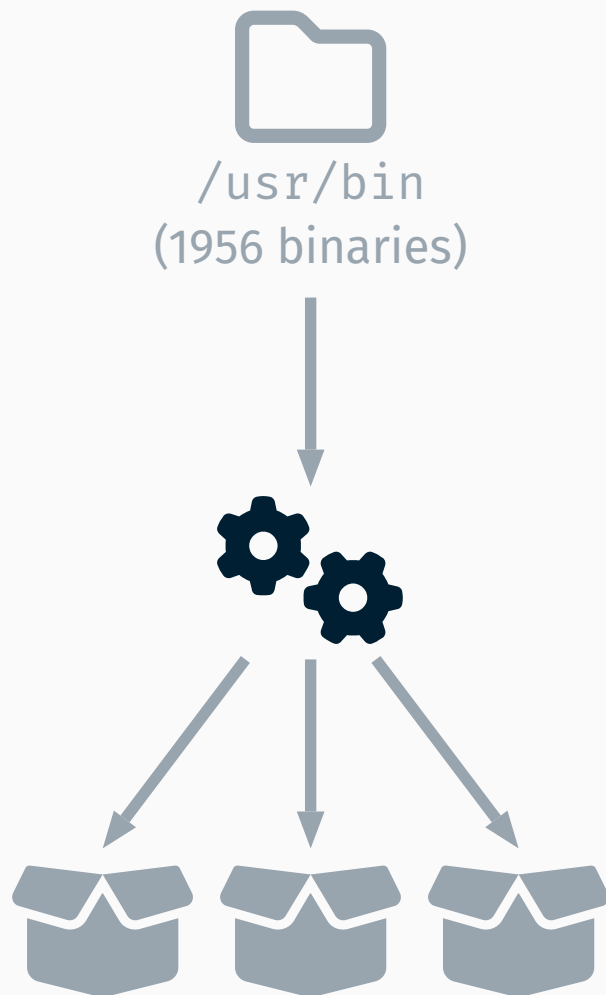
prepare_exec:

mov rdi, 0x1337

lea rsi, [rbp-0x20]

call execvp

ret



analyzed_code:

; ...

call fork

loop_head:

cmp rax, rcx

je prepare_exec

inc rcx

jmp loop_head

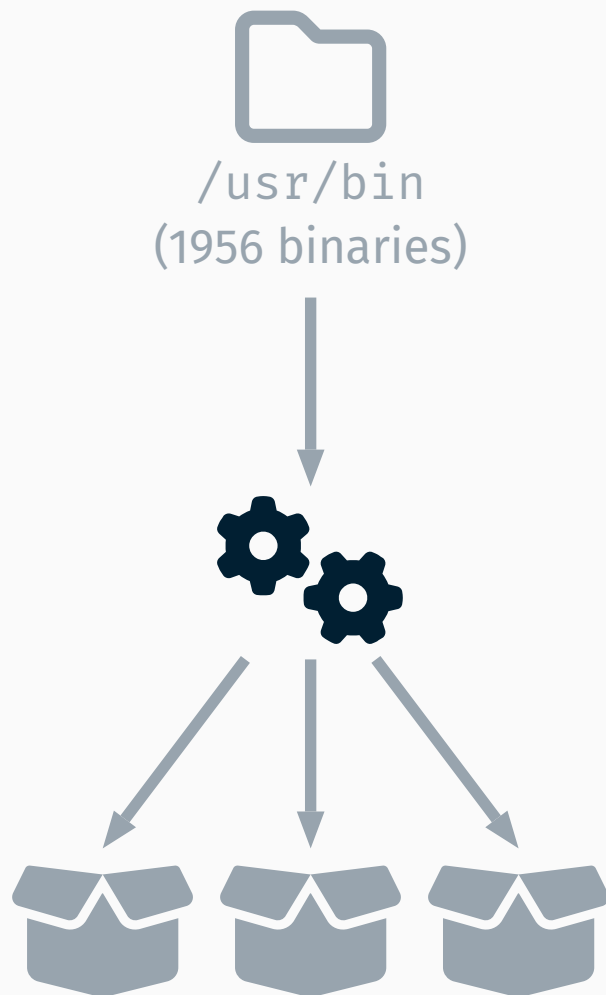
prepare_exec:

mov rdi, 0x1337

lea rsi, [rbp-0x20]

call execvp

ret



analyzed_code:

; ...

call fork

loop_head:

cmp rax, rcx

je prepare_exec

inc rcx

jmp loop_head

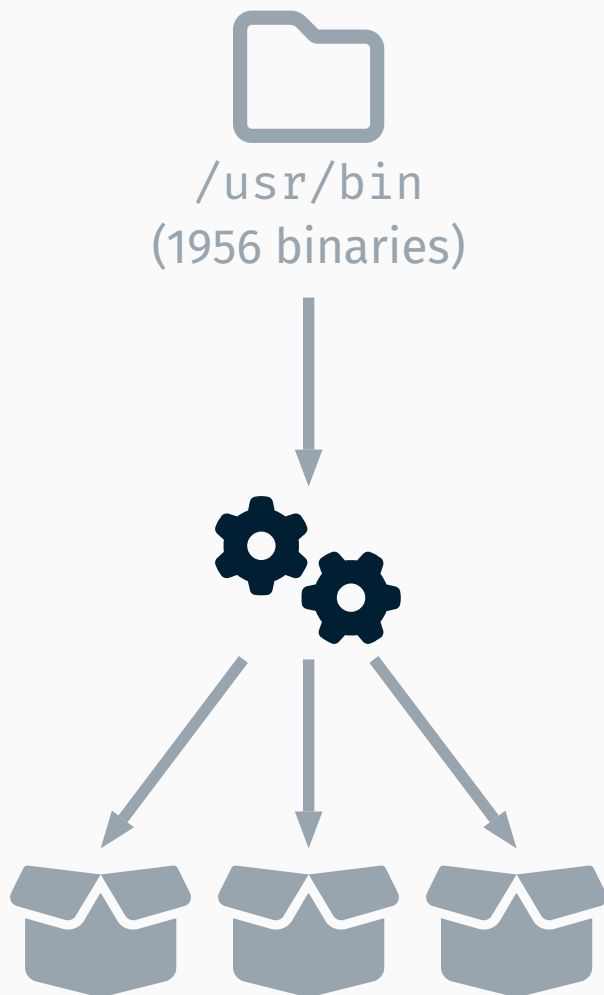
prepare_exec:

mov rdi, 0x1337

lea rsi, [rbp-0x20]

call execvp

ret



analyzed_code:

; ...

call fork

loop_head:

cmp rax, rcx

je prepare_exec

inc rcx

jmp loop_head

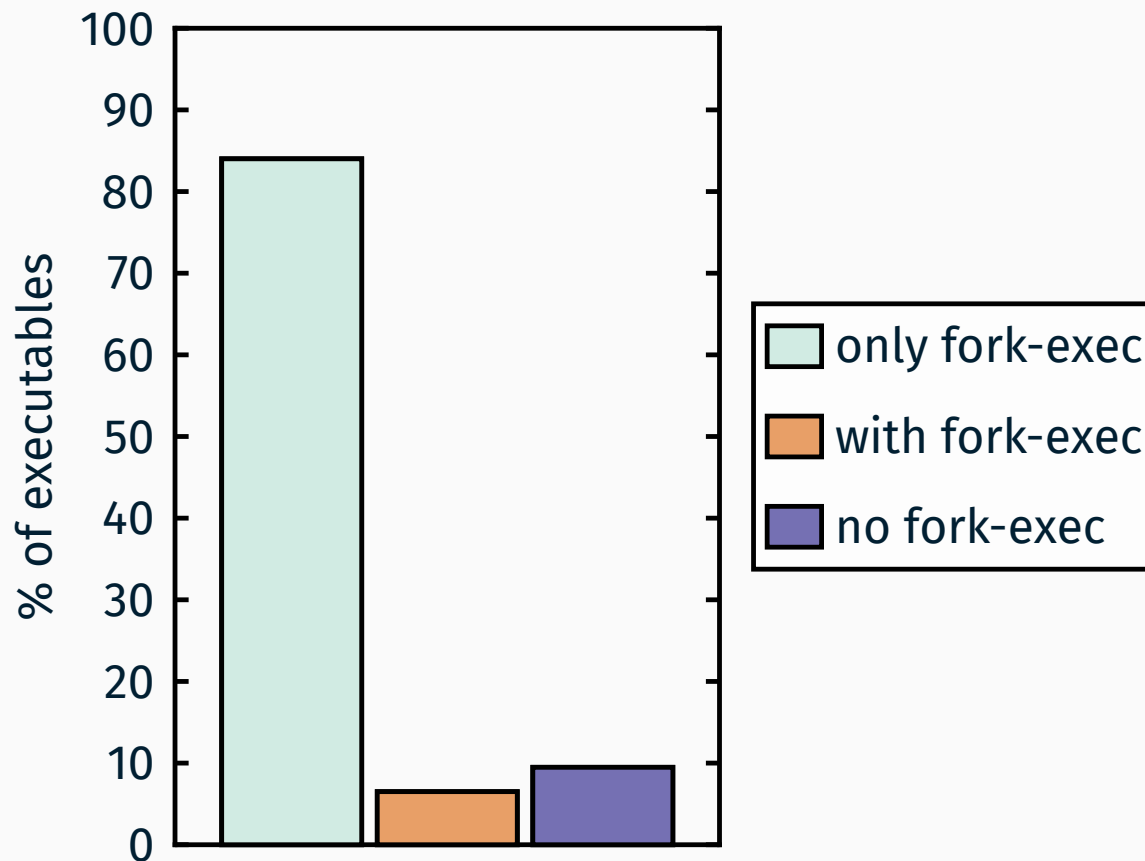
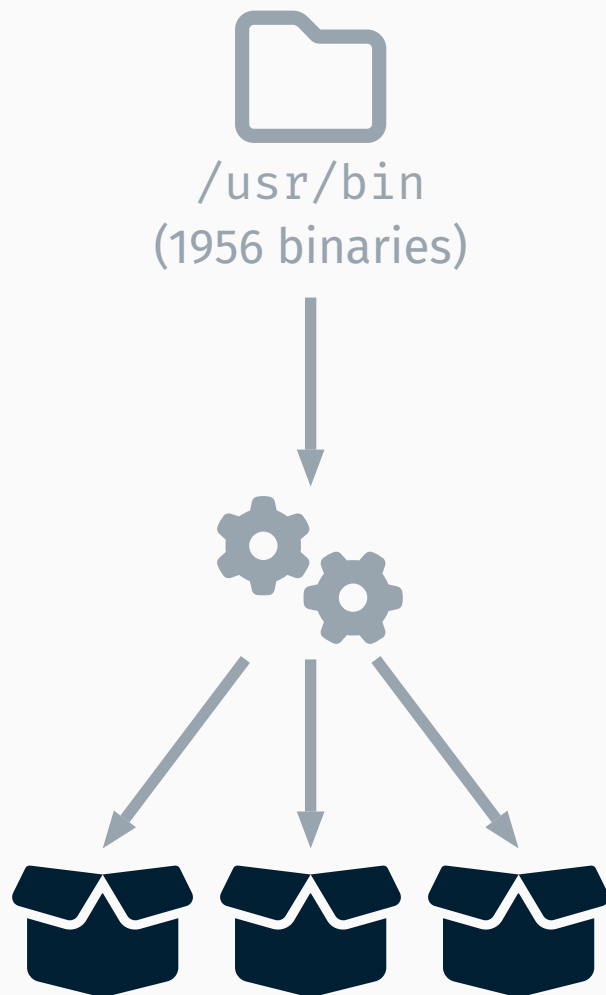
prepare_exec:

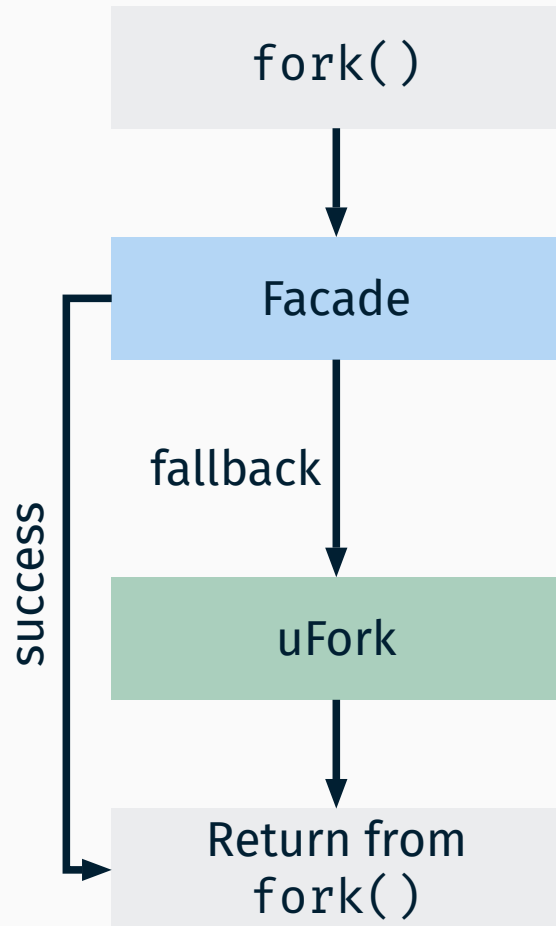
mov rdi, 0x1337

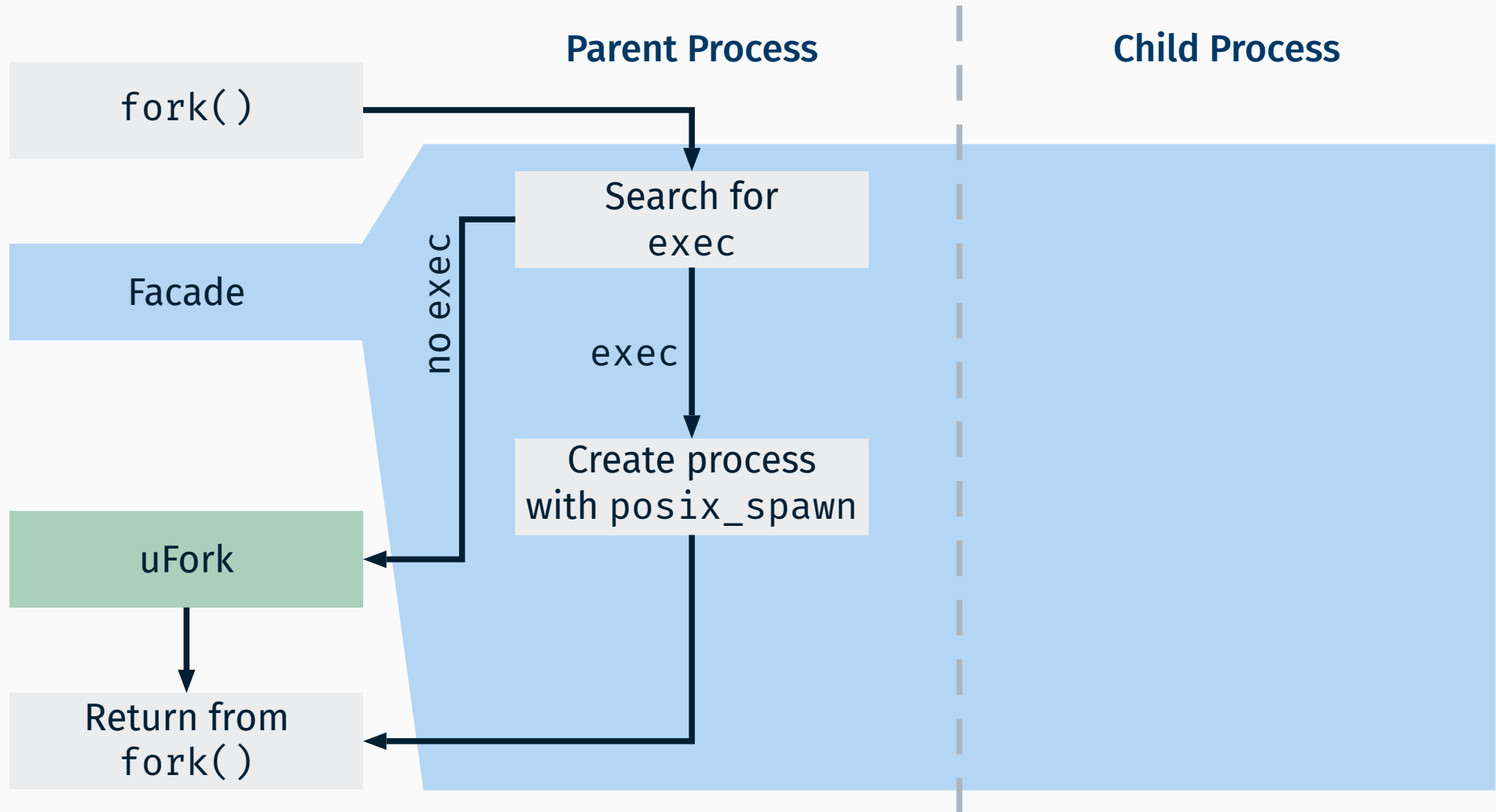
lea rsi, [rbp-0x20]

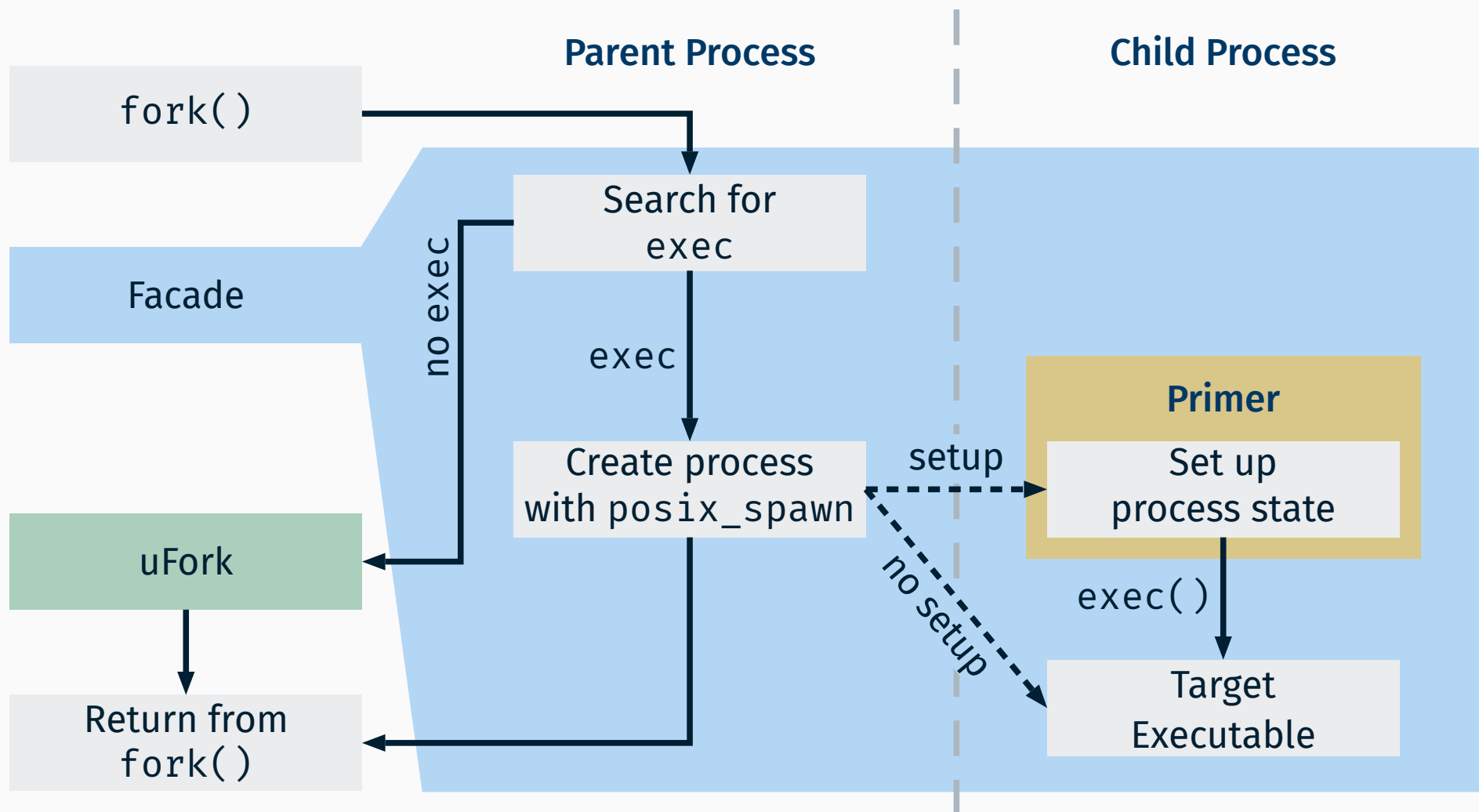
call execvp

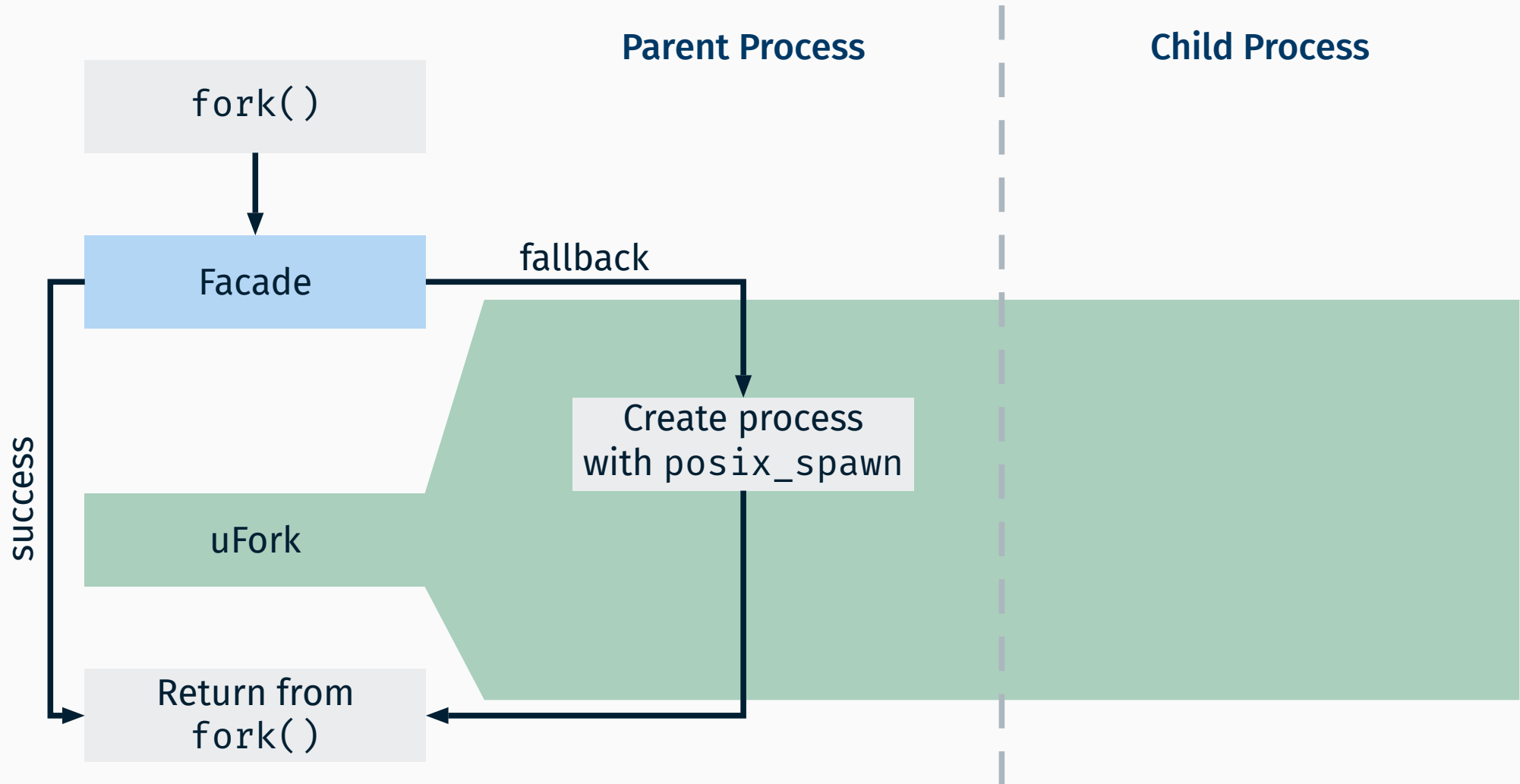
ret

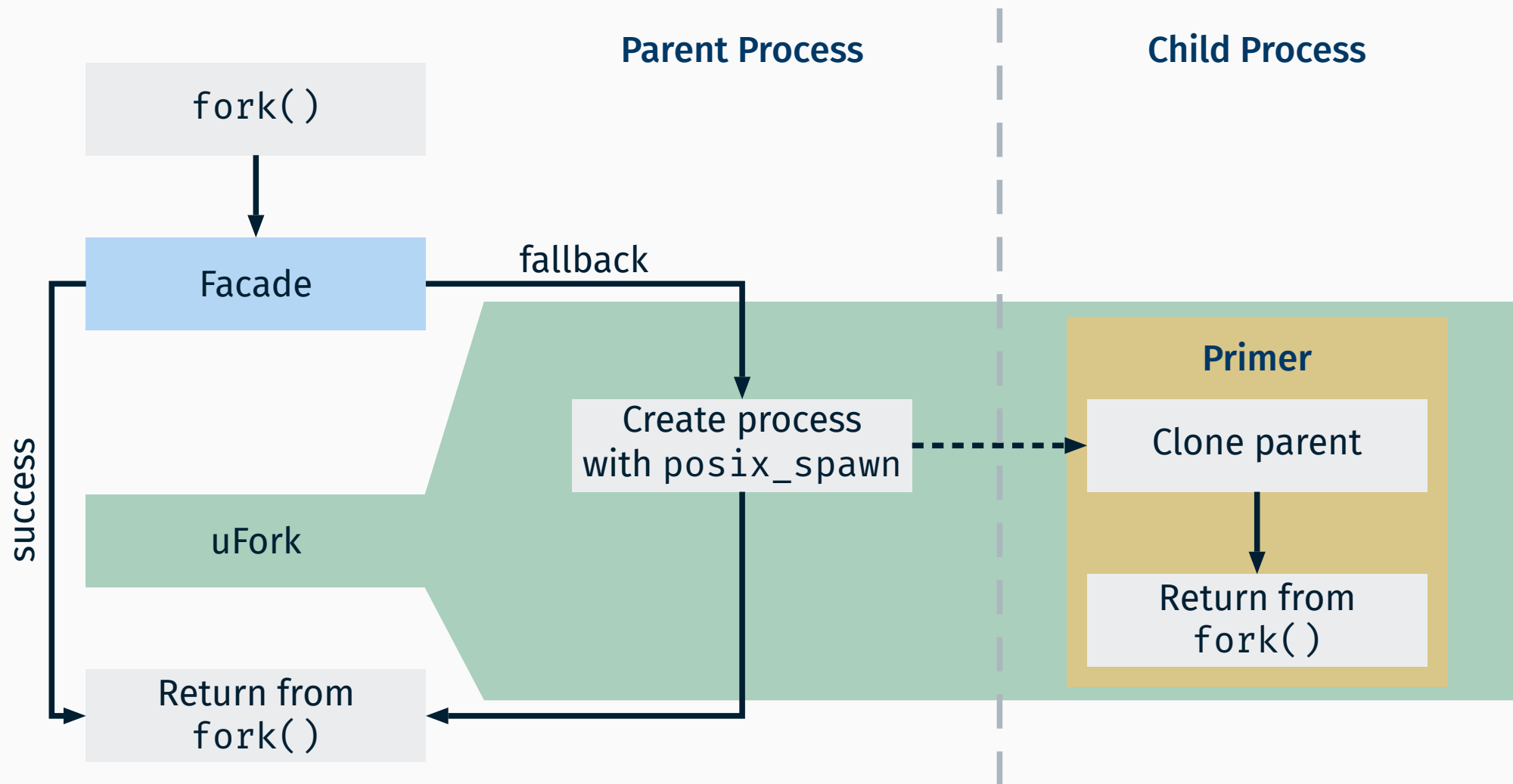














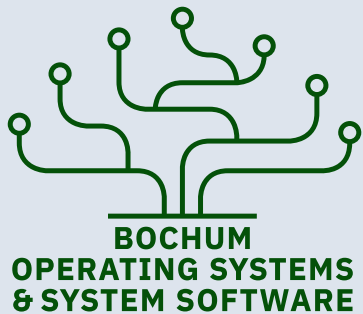
	fork	Spork
simple to implement	×	×
composes	×	✓
thread-safe	×	✓
secure API design	×	×
fast	×	(✓)
scalable	×	✓
no memory overcommit	×	✓

Spork

- liberates OS kernels from fork
- replaces fork without rewriting applications
- addresses most of fork's issues
- allows exploration of new OS concepts



Get in touch



<https://boss.rub.de>