Non-interference

So far: func. correctness ≈ integrity.

Keeping secrets: confidentiality.

"Open" problem.

Other APIs

Dedupe:

GetUsedBlocks() → 1.

Confid. not just about Read!
Non-interference spec

Intuition: B’s exec should not be affected by A.

A: \( W(0,10) \)  \( \rightarrow 0 \)  B: \( W(3,20) \)  \( \rightarrow 0 \)  B: \( R(2) \)  \( \rightarrow 10 \)

What data is confidential?

What data is observed/exposed?

Non-interference: prop. about 2 exec.
"2-safety prop."
Approach:
- Deterministic spec.
- Deterministic code.
- Observation func.

\[ O(\text{principal, state}) \rightarrow \text{subset of state that princ. can observe} \]

Spec. vs code obs.
\[ O(s_{\text{code}}) \leq O(s_{\text{spec}}) \]

Spec. obs.
\[ O(p, s_{\text{spec}}) \rightarrow p \text{ should be able to read this part of } s_{\text{spec}} \]
Implies confid. for \( p' \neq p \).

Code obs.
\[ O(p, s_{\text{code}}) \rightarrow \text{externally observable stack} \]
- Output to \( p \)'s terminal
- Packets sent to \( p \)'s computer.

\[ O(s) = O(s') \downarrow \]
\[ O(c) = O(c') \]
Proof of non-interference

Goal: Start out in states that have same spec. obs
→ Execute * → Observe same results at code level.

Assume:
\[ O(s_0) = O(s'_0) \]
\[ O(c_0) = O(c'_0) \]

Proof steps:
\[ O(s_2) = O(s'_2). \]

Goal:
\[ O(c_n) = O(c'_n). \]

What if spec \( O() \) is wrong?

What if code \( O() \) is wrong?

Restriction from prev. board.
Proving CertiKOS

**Principal**: process in OS.
Output terminal per process.

**Obs** \((p, s)\): virtual mem
- regs (if running)
- Saved regs (if not running)
- Output to terminal.

**API Issue**: PIDs

`fork() -> pid`
- global counter

**Proof Issue**: VM.

Write \((0, 5)\)

Spec \(P\):
- page table
- impl: phys pages

Spec Issue: local semantics

```
P: yield  
\[\text{\textbullet}}\]
\[\text{\textbullet}}\]
P: yield
\[\text{\textbullet}}\]
P: yield
\[\text{\textbullet}}\]
P: yield
\[\text{\textbullet}}\]
P: x
\[\text{\textbullet}}\]
```

```
P: yield
\[\text{\textbullet}}\]
P: x
\[\text{\textbullet}}\]
```

```
P: yield
\[\text{\textbullet}}\]
P: x
\[\text{\textbullet}}\]
```

```
P: yield
\[\text{\textbullet}}\]
P: x
\[\text{\textbullet}}\]
```

```
P: yield
\[\text{\textbullet}}\]
P: x
\[\text{\textbullet}}\]
```

```
P: yield
\[\text{\textbullet}}\]
P: x
\[\text{\textbullet}}\]
```

```
P: yield
\[\text{\textbullet}}\]
P: x
\[\text{\textbullet}}\]
```