Code Layout

15-411/15-611 Compiler Design

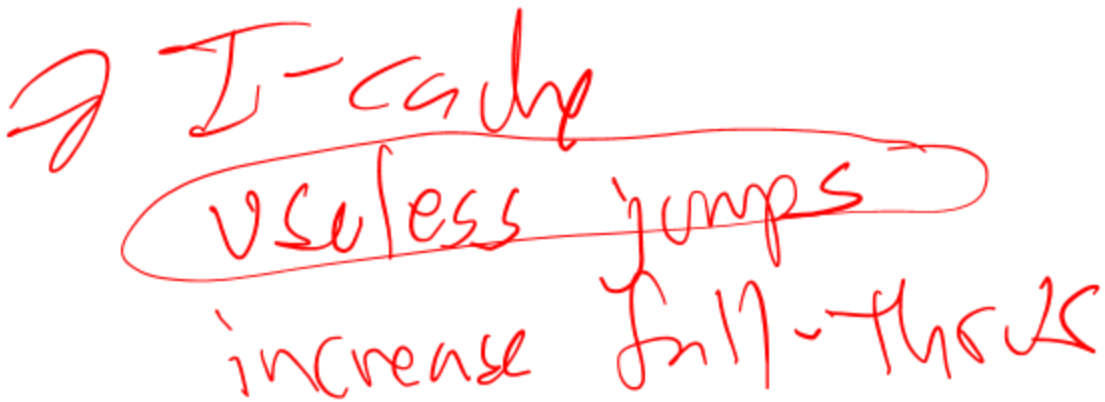
Seth Copen Goldstein

April 16, 2026

Today

- Why code layout matters
- Basic-block ordering and fall-through
- Static heuristics
- Hot/cold splitting
- Function placement and alignment
- Using Profile Data
- Example

Why code layout matters



Why code layout matters

- Architecture
  - I-Cache
  - Branch predictor
  - Fetch
- Program Behavior
  - Hot v. Cold
  - Traces

Today

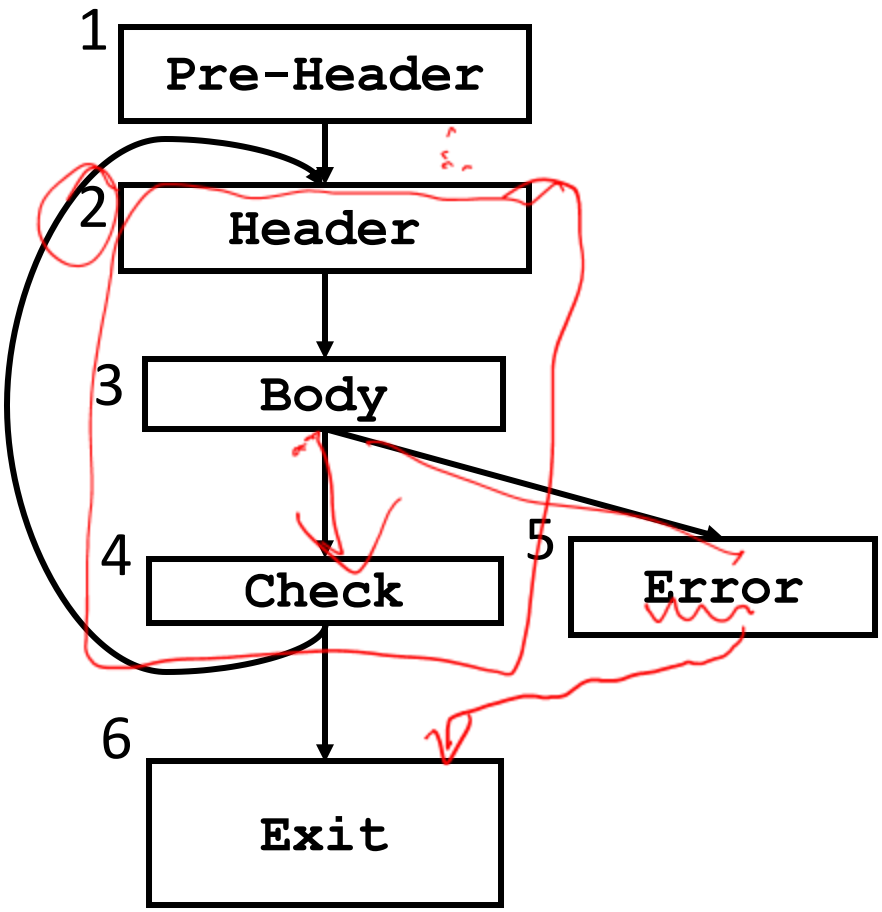
- Code Layout
- Alignment

The Layout Problem

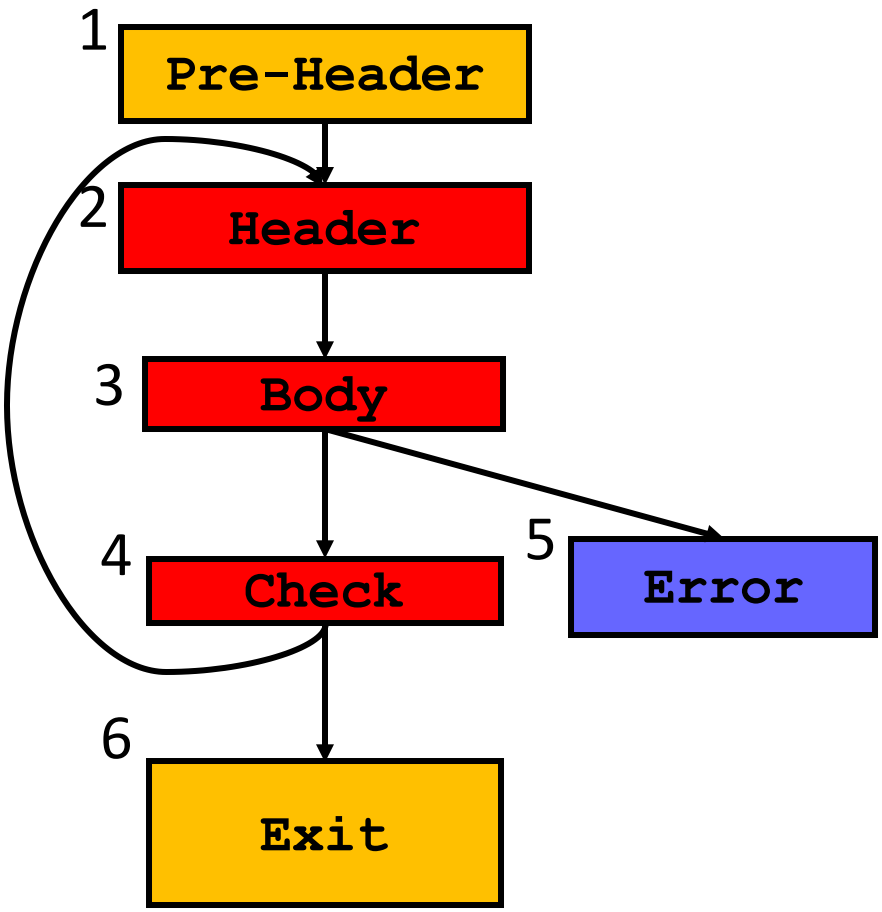
- Given:
  - CFG
  - Optional: Profile Data
- Determine:
  - Order of Blocks
  - Alignment of Blocks
- Plan:
  - Increase fall-through cases
  - Keep hot blocks/traces together
  - Make hot targets cache aligned

Linearize the CFG so the hot path is cheap

The Layout Problem



The Layout Problem



Metrics

- Fewer taken branches on hot paths
- Fewer mispredictions on hot paths
- Smaller hot instruction footprint
- Better locality for frequently adjacent blocks
- Keep cold code out of the middle of hot code

- Balance: locality, prediction, code-size

Branch Inversion

- It isn't just about moving code
- Branch Inversion
  - Keep semantics
  - Keep fall-through on likely path

Before

```
if likely_condition
    goto Hot
Cold:

Hot:
```

After

```
if !likely_condition
    goto Cold
Hot:



Cold:
```

No Profile?

- Profile data is best
- No profile, can still ”guess”
  - Loops
  - Function calls
  - Assertions
  - Error handling

Loops are usually Hot

- Backedges are often taken many times
- Loop bodies are usually hotter than exits
- Keep the common loop path compact
- Prefer the loop body and backedge path to look sequential when possible

Errors/Assertions/Cleanup

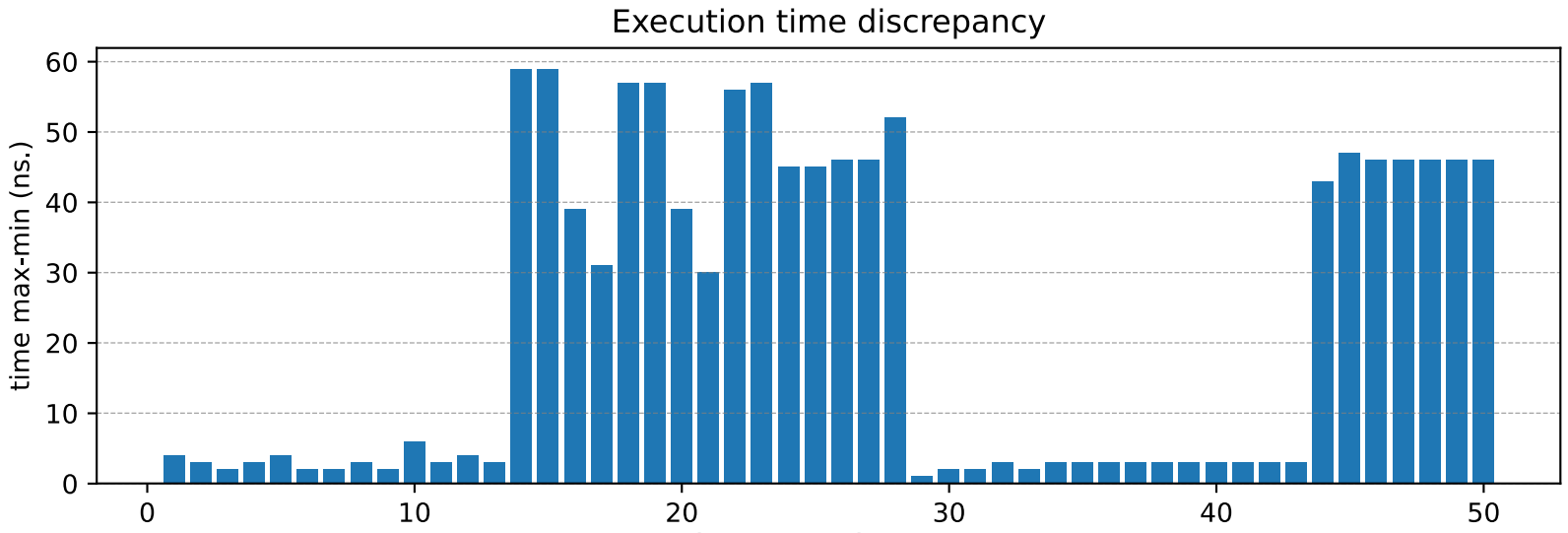
- Rare case gets extra jump
- Really rare case gets put in cold region
- How to detect without profile data?

Function Placement

- Hot functions should be together
- Interacting functions should be together
- (First focus on basic blocks)

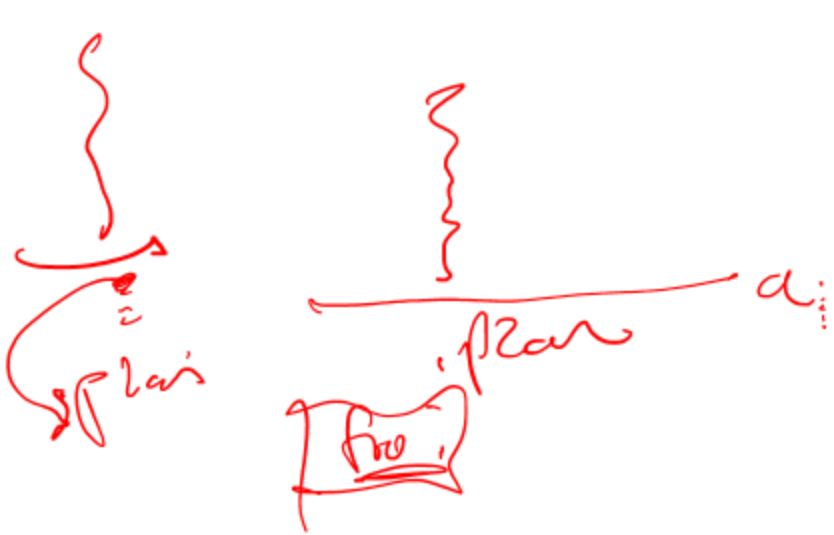
Alignment

- Surprisingly important effect
- Micro-architecture dependent
- However, beware code bloat



www.bazhenov.me/posts/2024-02-performance-roulette/

- Where?
  - Function Entry
  - Hot loop headers
  - Hot branch targets
- How: **`.p2align`**
  - **`.p2align 4`**
    align to 16 byte boundary
  - **`.p2align 5,,7`**
    align to 32 byte boundary if can do so with fewer than 7 bytes of filler



Profile Data

- Block counts
- Edge Frequencies
- Can help identify
  - Hot targets (for alignment)
  - Hot blocks (for placement)
  - Common paths (for fall through)
  - Hot Functions
  - …

Summary

- Make common case sequential & Close
- Convert to fall-through if necessary
- Hot blocks & Hot functions should be close together
- Rare paths go far away
- Use profile data if possible
- Use alignment with care

End of Semester Admin

- Tue 4/21 No Lecture, Office hours
- Wed 4/22 Checkpoint for Friday's Assessement
  Form will be online
- Thu 4/23 Final wrap-up lecture
- Fri 4/24 In recitation assessment
  YOU MUST GO TO ASSIGNED ROOM
- Sat 4/25 Compiler
- Sat 5/2 Report (grading will be tight)

Scoreboard?