

# Advanced Rust 2026 - Lab 6: Systems Interfaces and Robust I/O

Lukáš Hozda

Spring 2026

## Lab Goals

1. Design robust wrappers around system interfaces.
2. Map OS-level failures into domain errors.
3. Prepare for homework pair A6.

## Time Plan (90 min)

1. 15 min - system boundary recap
2. 25 min - exercise 1 (process execution wrapper)
3. 25 min - exercise 2 (line-oriented log processor)
4. 15 min - exercise 3 (timeout/error policy)
5. 10 min - review

## Exercise 1: Command Runner

Implement helper:

```
fn run_checked(program: &str, args: &[&str]) → Result<String, RunError>
```

Requirements:

1. Capture stdout.
2. Surface non-zero exit clearly.
3. No panic on process errors.

## Exercise 2: Buffered Log Processing

Read large text file line by line and compute:

1. count of valid records
2. count of malformed records
3. aggregate numeric field sum

## Exercise 3: Failure Policy

Define explicit policy for:

1. unreadable file
2. invalid line format
3. integer overflow

Compare strict-fail-fast and skip-invalid policies.