Advanced Rust (2026): Systems Interfaces

Lecture 6

Lukáš Hozda

Spring 2026

MFF CUNI

Systems Thinking

Scope Of Systems Rust

- OS interaction: files, processes, sockets, signals.
- Data representation across boundaries.
- Failure-aware control flow.

Error Mapping

- Preserve context when crossing layers.
- Convert low-level errors to domain-level meanings.

```
fn read_config(path: &std::path::Path) -> Result<String, String> {
    std::fs::read_to_string(path)
        .map_err(|e| format!("cannot read {}: {e}", path.display()))
}
```

File Descriptors And Ownership

- `File` owns descriptor and closes on drop.
- `AsRawFd` borrows descriptor.
- `IntoRawFd` transfers ownership explicitly.

Process Pipelines

- Build commands as data.
- Capture stdout/stderr separately.
- Define timeout and failure policy.

```rust
let output = std::process::Command::new("rustc")
    .arg("--version")
    .output()
    .expect("failed to execute rustc");
println!("{}", String::from_utf8_lossy(&output.stdout));
```

Buffered I/O

- Syscalls are expensive.
- `BufRead` / `BufWriter` reduce overhead.
- Flush strategy is part of correctness for interactive tools.

Binary Layout Concerns

- `repr(C)` only when ABI compatibility is required.
- Endianness and alignment must be explicit in protocols.

Time And Clocks

- Monotonic clock for intervals/timeouts.
- Wall clock for user-facing timestamps.
- Never mix semantics accidentally.

Signal And Shutdown Strategy

- Cancellation path should preserve invariants.
- Graceful shutdown is design, not add-on.
- Keep shutdown idempotent.

Resource Lifetime Design

- RAII for cleanup.
- Explicit ownership transfer at boundaries.
- Avoid hidden global state.

Observability

- Structured logs over ad-hoc prints.
- Include correlation ids in concurrent tools.
- Capture enough context for post-mortem analysis.

Practical Rule

- Prefer straightforward design + clear invariants.
- Optimize only after measurement.
- Systems code quality = correctness + diagnosability.