ENHANCED OBSERVABILITY AND MONITORING IN XV6

Kernel-Level Flight Recorder & Unified System Monitoring Dashboard

TANVIR BIN ZAHID 2131717042

OVERVIEW

Modern operating systems rely on observability for debugging, performance tuning, and fault diagnosis. xv6, a minimalist teaching OS, lacks these capabilities. This project enhances xv6 with two integrated features:

1. **Kernel-Level Flight Recorder (Black Box Recorder)**
2. **Unified System Monitoring Dashboard (sysdash)**

FEATURE 1 — KERNEL-LEVEL FLIGHT RECORDER (BLACK BOX RECORDER)

OBJECTIVE

Implement a **circular kernel log buffer** that records the most recent OS events.
This allows post-crash analysis and real-time debugging.

MOTIVATION

xv6 provides no persistent system logs, making kernel crashes difficult to diagnose.
A flight recorder captures essential events leading to a fault, similar to aircraft black boxes and Linux kernel ring buffers.

KEY CAPABILITIES

- Log system calls
- Log context switches
- Log traps and timer interrupts
- Log process creation (fork) and termination (exit)
- Automatically dump the last N entries when the kernel crashes (panic())

DESIGN OVERVIEW

A ring buffer stores entries: timestamp, event_type, pid, extra. Hooks added in syscall(), trap(), sched(), fork(), exit(). A user tool 'klogs' prints recent logs.

EXPECTED OUTCOMES

- Significant improvement in debugging ability
- Insight into scheduling and process behavior
- A realistic model of OS diagnostic systems

FEATURE 2 — UNIFIED SYSTEM MONITORING DASHBOARD (SYSDASH)

This feature merges two ideas:
(1) enhanced process statistics
(2) virtual CPU temperature & load monitoring into **one integrated dashboard**.

OBJECTIVE

Provide a live OS dashboard similar to Linux top/htop.)

MOTIVATION

xv6 currently lacks visibility into its process table and CPU behavior.
A dashboard improves understanding of scheduling, resource use, and system state.

KEY COMPONENTS

1) SYSTEM CALLS

Two new system calls will expose kernel data:
getprocstats(struct pstat *buf, int max);
getkernelstats(struct kstat *ks);

2) VIRTUAL CPU TEMPERATURE

A synthetic temperature model updates on every timer tick:

- Busy tick → temperature increases
- Idle tick → temperature decreases
- Range: 35°C – 95°C
- Warning thresholds (e.g., >80°C)

3) SYSDASH USER PROGRAM

Expected displays a formatted table:

```
================== xv6 System Dashboard ==================
Uptime: 2134 ticks      CPU Temp: 57°C      Load: 63%
----------------------------------------------------------
PID   Name      State     CPU%    Mem    CtxSw   Runtime
...
----------------------------------------------------------
Warnings:
- CPU heating up!
```

EXPECTED OUTCOMES

- Better visibility into process behavior
- Real-time system analysis
- Stronger practical understanding of OS resource management

PROPOSAL

We propose to extend xv6 with **two integrated, high-impact system monitoring features** that significantly enhance observability:

1. **Kernel-Level Flight Recorder**
   This will provide structured, chronological visibility into system events, making debugging and teaching substantially more effective.

2. **Unified System Monitoring Dashboard**
   This provides a real-time, interactive view of processes, CPU behavior, virtual thermal conditions, and system load, transforming xv6 into a more realistic and educational environment.

These features are strongly aligned with real-world operating system design and debugging needs. They are technically achievable within the project timeline and will produce a polished, visually demonstrable outcome.