Real-Time Operating Systems (RTOS)

- jadi.net
- telegram: jadivarlog
- instagram: jadijadinet
- youtube: jadimirmirani
- mastodon: @jadi@mastodon.com
- nostr: nprofile1qqsv9zhvu4327t2ax0newawjnlnqycpru40xgxluq0sn3vsxdmv3zwsz8pcwz
- ...
- jadi.ir/links

Introduction

Real-Time Operating Systems (RTOS)

- **What is an RTOS?**
  - Designed for **deterministic behavior**
  - Handles tasks with strict timing constraints
- Applications:
  - IoT, robotics, aerospace, automotive
  - My Impression
- **Focus of the Talk:** RTOS in Linux

What is an RTOS?

Key Characteristics

- **Determinism:** Consistent response times
- **Low Latency:** Immediate response to critical tasks
- **Scheduling:** Priority-based task management

Hard vs. Soft Real-Time

Type	Example	Timing Requirement
Hard	Pacemakers, Aerospace	Must meet deadlines
Soft	Video Streaming	Best effort, tolerable delays

Real-Time in Linux

Evolution of Linux

- Initially designed as a **general-purpose OS**
- Support for real-time workloads through:
  - **PREEMPT_RT [patch]**
  - RT Schedulers

Real-Time Scheduling Policies

- **SCHED_FIFO:** First in, first out
- **SCHED_RR:** Round robin
- **SCHED_DEADLINE:** Earliest deadline first

Methods for Real-Time in Linux

Achieving Real-Time Performance

1. **Kernel Configuration:**
   - Apply PREEMPT_RT patch / Use 6.12+ Kernels
   - Enable real-time settings
   - Code for RT

Code Example

Real-Time LINUX

```c
#include <sched.h>
#include <stdio.h>
#include <stdlib.h>
#include <errno.h>
#include <unistd.h>

int main() {
    struct sched_param param;
    param.sched_priority = 80;

    if (sched_setscheduler(0, SCHED_FIFO, &param) == -1) {
        perror("sched_setscheduler failed");
        exit(EXIT_FAILURE);
    }

    printf("Running as a real-time task with SCHED_FIFO\n");
    while (1) {
        printf("I'm RT!\n");
        sleep(1);
    }
    return 0;
}
```

Conclusion

Summary

- RTOS ensures deterministic and low-latency task management.
- Linux, with PREEMPT_RT and tuning, supports real-time workloads.
- Example code demonstrates real-time scheduling in action.

Questions?

- Feel free to ask!
- Suggestion to go deeper?
    - wiki.linuxfoundation.org/realtime/start
    - kernel/sched
    - kernel/sched/rt.c
    - kernel/sched/core.c ~ 5951