2025 제 10 회 초소형위성 워크샵

Presentation [5-6]

Development and Validation of Reliable Kernel with Real-Time Multitasking Support

박 주 찬

한남대학교

고신뢰성 실시간 멀티태스킹 지원 커널 개발 및 검증 LIG Nex1

2025년 제10회 초소형위성 워크샵

2025.05.30

김태한

한남대학교 시스템소프트웨어 연구실

SSLab

목차

SSLab

1 연구 배경

우주 분야 RTOS 동향

SSLab

RTOS(Real-Time Operating System) 최신 동향

RTOS	유형	주요 사용 사례	특징	비고
VxWorks	상업용	NASA 큐리오시티, 퍼서비어런스, 인사이트	고신뢰 상업 RTOS, 미션 검증 완료	대형 미션 중심
RTEMS	오픈소스	ESA 위성, NASA CubeSat	ESA 표준, SMP 지원, 오픈소스	CubeSat 확산
FreeRTOS	오픈소스	CubeSat OBC, 소형 탑재체	초경량 커널, 빠른 개발	실험 위성 최적
Zephyr	오픈소스	민간 CubeSat (Space Cubics 등)	IoT 기반 모듈식 구조, 네트워크 내장	신흥 활용
INTEGRITY-178B	상업용	NASA 오리온 유인 우주선	DO-178 인증, 파티셔닝 RTOS	유인/고신뢰 임무

우주 분야 RTOS 동향

SSLab

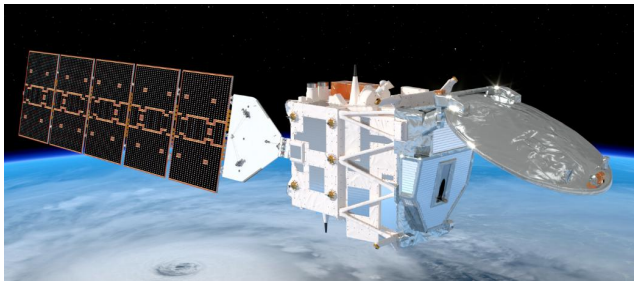
- 오픈소스 RTOS 활용 확대
  - **RTEMS**: ESA 표준 → CubeSat, LEON, RISC-V 기반 확장 중
  - **FreeRTOS**: CubeSat OBC, 초경량 미션용
  - **Zephyr**: 민간 스타트업 중심 시도, IoT 생태계 확장

- 상업 RTOS (VxWorks, INTEGRITY-178B)
  - 대형 미션, 고신뢰/인증 요구 임무에서 지속 활용

- 하이브리드 아키텍처 증가
  - XtratuM 하이퍼바이저 사용: RTEMS+Linux 병행 (OneWeb, SWOT 위성 등)
  - 리눅스 기반 실험적 CubeSat도 등장 (SpaceX, NASA Ingenuity 등)

RTEMS의 우주산업 활용

위성 미션 (2020년 이후)
- **EarthCARE (ESA, 2024)**
- **PLATiNO (ASI, 2023)**

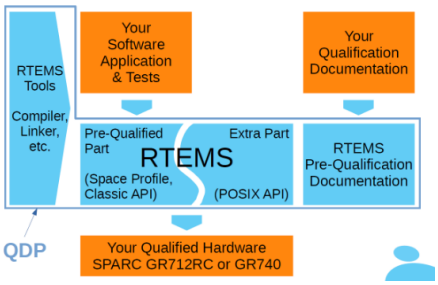


로버 미션
- **Perseverance Rover (NASA, 2021)**

RTEMS QDP

SSLab

Qualification Data Package



구분	일반 RTEMS 배포판	QDP 버전 (ESA 배포)
문서/자료	공식/비공식 문서 풍부	QDP 전용 자료 희박
커뮤니티 지원	활발, 문제 해결 빠름	한정적, 직접 분석 필요
호환성 문제	적음	일반 RTEMS 기능 일부 미지원
사용자 수	상대적으로 많음	매우 적음

2 연구 목적

연구 필요성

SSLab

연구의 필요성

- 일반 배포판과 QDP 버전의 **기능, 빌드 호환성 차이 분석**
- 자료 부족으로 인한 **초기 사용자 진입 장벽 해소**

연구 목표

SSLab

- **최종 연구 목표**
  - 고신뢰성 실시간 멀티태스킹 지원 커널 개발 및 검증

- **세부 연구 목표**
  - **실시간성 보장을 위한 고신뢰성 커널 개발**
    - 실시간 성능을 보장하는 커널 개발 → 위성 시스템 신뢰성 및 안정성 확보
  - **멀티태스킹 지원 및 스케줄 관리 기능**
    - 효율적인 작업 전환, 우선순위 관리, 리소스 할당을 포함한 멀티태스킹 기능 구현
    - 실시간 제약 조건을 처리할 수 있는 스케줄링 시스템 설계 → 작업 간의 효율적인 조정 지원
    - RSA(RTEMS System Administrator) 개발을 통한 스케줄링 최적화

3 RTEMS 빌드 방식 변화

RTEMS 빌드 분기점

SSLab

RTEMS 버전에 따른 빌드 방식 변화

GNU Make
(Autotools 기반)

Waf
(Python 기반)

이유	설명
빌드 속도 개선	빌드 시간 대폭 단축
복잡도 감소	빌드 스크립트 간소화
유지보수 용이	Python 코드 기반 관리
다중 BSP 지원	여러 BSP 동시 빌드 가능
품질 인증 대응	빌드 과정 명확화

RTEMS 빌드 방식 차이

SSLab

비교 항목	기존 GNU Make (Autotools 기반)	새로운 Waf (Python 기반)
구성 방식	Autoconf의 configure 스크립트가 환경 검사 후 다수의 Makefile 생성. Make가 규칙 해석.	Python 기반 wscript로 빌드 규칙 정의. **Makefile 생성 없이** 곧바로 빌드 실행.
빌드 기술 언어	M4 매크로(autoconf), Makefile DSL 등 **여러 전용 문법** 사용.	**Python**으로 빌드 스크립트 작성 (별도 DSL 없음).
종속성 관리	규칙에 따라 수동 관리 또는 컴파일러 도움으로 .d 파일 생성. 누락 시 재빌드 문제 발생 가능.	**내장 스캐너**로 소스의 include/의존 파일 자동 추적. 변경 파일만 **선별 재컴파일**.
병렬화 및 성능	make -j로 병렬 빌드 가능하나 configure 단계 등 **오버헤드 큼**. 대규모 프로젝트에서 빌드 시간 지연.	**병렬 작업 최적화** 및 가벼운 설정 단계로 **전체 빌드 속도 향상** (빌드 시간 **대폭 단축**).
확장성/유연성	설정 옵션 추가 시 다수의 매크로/Makefile 수정 필요. **다중 BSP** 대응 논리 복잡.	Python 코드로 **유연하게 조건 처리** 가능. **YAML/INI 스펙** 기반으로 다중 BSP 구성 **간소화**.
필요 조건	Autoconf, Automake, GNU Make 등 **외부 툴체인 요구**.	Python 3만 필요 (Waf 스크립트는 프로젝트에 포함되거나 다운로드).

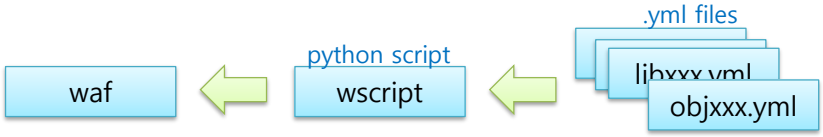
4 RTEMS QDP 빌드

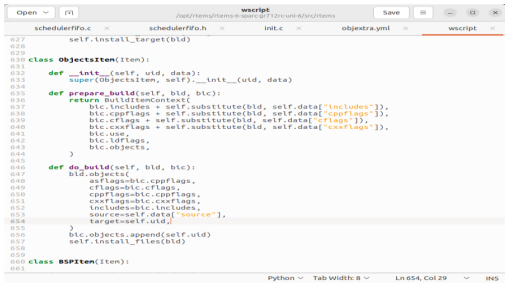
RTEMS 빌드 시스템 분석

SSLab

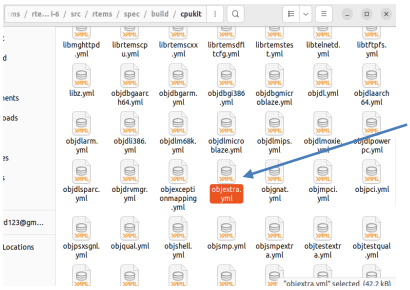
RTEMS 빌드 시스템 분석

- RTEMS 빌드 도구: make → waf & RSB(RTEMS Source Builder)
- RTEMS 6.x 버전에서는 애플리케이션 및 사용자 코드를 waf로 빌드
- 소스 파일 추가: 각 기능에 해당하는 .yml에 파일경로를 추가해서 구성 변경





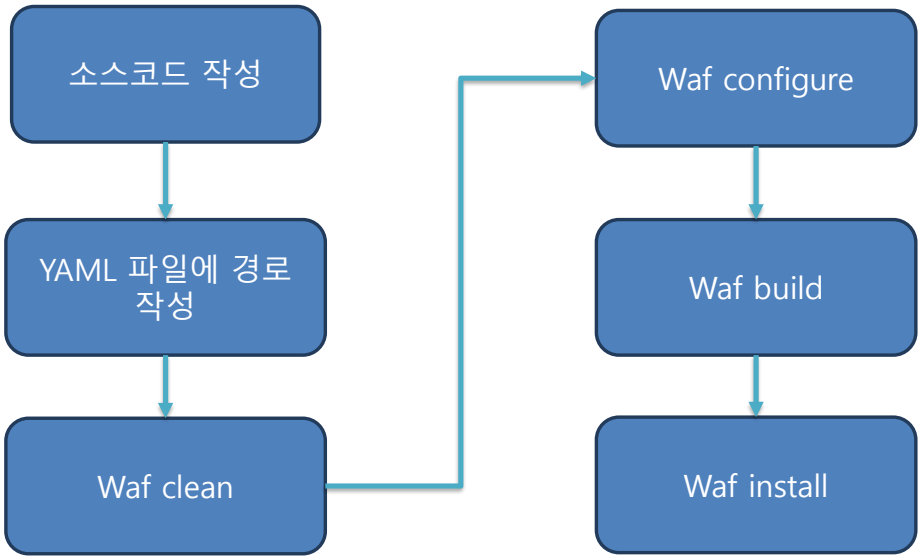
<wscript>



<.yml files>

RTEMS QDP 빌드

SSLab



RTEMS QDP 빌드 과정(1)

SSLab

- 소스 코드 작성

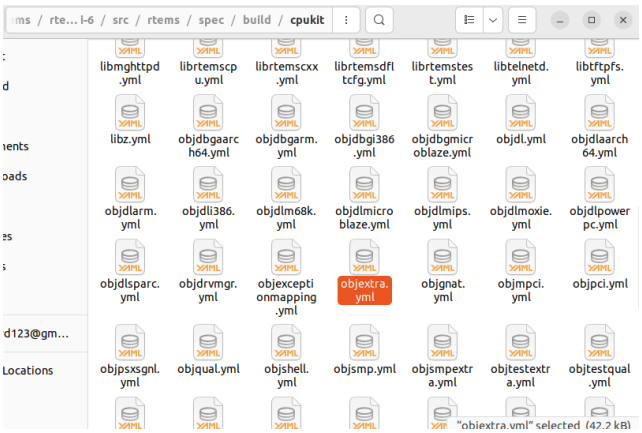
schedulerfifo.c | schedulerfifo.h | init.c | scheduler.h

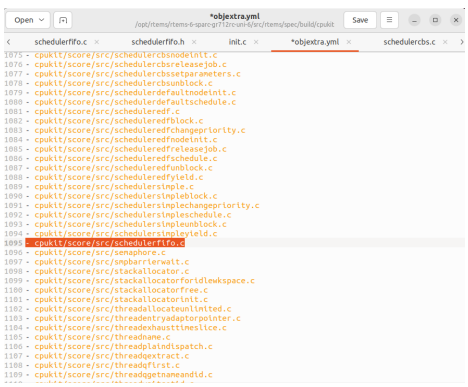
```
#ifndef _RTEMS_SCORE_SCHEDULERFIFO_H
#define _RTEMS_SCORE_SCHEDULERFIFO_H

#include <rtems/score/scheduler.h>
#include <rtems/score/chainimpl.h>

#ifdef __cplusplus
extern "C" {
#endif

typedef struct {
    Scheduler_Control Base;
    Chain_Control ready_queue;
} Scheduler_FIFO_Control;

/* Scheduler Operations */
void _Scheduler_FIFO_Initialize(const Scheduler_Control *scheduler);
Thread_Control *_Scheduler_FIFO_Schedule(const Scheduler_Control *scheduler);
void _Scheduler_FIFO_Yield(const Scheduler_Control *scheduler, Thread_Control *thread);
void _Scheduler_FIFO_Block(const Scheduler_Control *scheduler, Thread_Control *thread);
void _Scheduler_FIFO_Unblock(const Scheduler_Control *scheduler, Thread_Control *thread);
void _Scheduler_FIFO_Enqueue(const Scheduler_Control *scheduler, Thread_Control *thread);
void _Scheduler_FIFO_Enqueue_first(const Scheduler_Control *scheduler, Thread_Control *thread);
void _Scheduler_FIFO_Extract(const Scheduler_Control *scheduler, Thread_Control *thread);

/* External Scheduler Definition */
extern const Scheduler_Control _Scheduler_FIFO;
extern Scheduler_FIFO_Control _Scheduler_FIFO_Instance;

#ifdef __cplusplus
}
#endif

#endif /* _RTEMS_SCORE_SCHEDULERFIFO_H */
```

RTEMS QDP 빌드 과정(2)

SSLab

- YAML 파일에 경로 작성





RTEMS QDP 빌드 과정(3)

SSLab

- waf clean, waf configure

```
ditto@RTEMS:/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems$ ./waf clean
'clean' finished successfully (0.060s)
err: /bin/sh: 1: git: not found
'clean_sparc/gr712rc-extra' finished successfully (0.154s)
ditto@RTEMS:/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems$ ./waf configure --
prefix=/opt/rtems/rtems-6-sparc-gr712rc-uni-6
Setting top to                           : /opt/rtems/rtems-6-sparc-gr712rc-uni-
6/src/rtems
Setting out to                           : /opt/rtems/rtems-6-sparc-gr712rc-uni-
6/src/rtems/build
Inherit options from 'sparc/gr712rc'     : sparc/gr712rc-extra
Configure board support package (BSP)    : sparc/gr712rc-extra
Checking for program 'sparc-rtems6-gcc'  : /opt/rtems/rtems-6-sparc-gr712rc-uni-
6/bin/sparc-rtems6-gcc
Checking for program 'sparc-rtems6-g++'  : /opt/rtems/rtems-6-sparc-gr712rc-uni-
6/bin/sparc-rtems6-g++
Checking for program 'sparc-rtems6-ar'   : /opt/rtems/rtems-6-sparc-gr712rc-uni-
6/bin/sparc-rtems6-ar
Checking for program 'sparc-rtems6-ld'   : /opt/rtems/rtems-6-sparc-gr712rc-uni-
6/bin/sparc-rtems6-ld
Checking for program 'ar'                : /opt/rtems/rtems-6-sparc-gr712rc-uni-
6/bin/sparc-rtems6-ar
Checking for program 'g++, c++'          : /opt/rtems/rtems-6-sparc-gr712rc-uni-
```

RTEMS QDP 빌드 과정(4)

SSLab

- waf build

```
ditto@RTEMS:/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems$ ./waf build
Waf: Entering directory `/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems/build'
Waf: Leaving directory `/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems/build'
'build' finished successfully (0.035s)
Waf: Entering directory `/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems/build/
sparc/gr712rc-extra'
err: /bin/sh: 1: git: not found
[   1/4478] Compiling bsps/shared/rtems-version.c
[   2/4478] Compiling bsps/shared/dev/i2c/spi-fram-fm25l256.c
[   3/4478] Compiling bsps/shared/dev/rtc/icm7170_reg.c
[   4/4478] Compiling bsps/shared/dev/display/disp_hcms29xx.c
[   5/4478] Compiling bsps/shared/ofw/ofw.c
[   6/4478] Compiling bsps/shared/dev/serial/ns16550.c
[   7/4478] Compiling bsps/shared/dev/rtc/ds1375.c
[   8/4478] Compiling bsps/shared/dev/i2c/i2c-sc620.c
[   9/4478] Compiling bsps/shared/start/bootcard.c
[  10/4478] Compiling bsps/shared/dev/ide/ata_util.c
[  11/4478] Compiling bsps/shared/dev/rtc/icm7170_reg2.c
[  12/4478] Compiling bsps/shared/freebsd/sys/arm/ti/ti_pinmux.c
[  13/4478] Compiling bsps/shared/dev/serial/ns16550-context.c
[  14/4478] Compiling bsps/shared/dev/rtc/m48t08_reg8.c
[  15/4478] Compiling bsps/shared/dev/i2c/spi-memdrv.c
[  16/4478] Compiling bsps/shared/dev/rtc/icm7170_reg4.c
```

RTEMS QDP 빌드 과정(5)

SSLab

- **waf install**

```
parc/gr712rc-extra'
'build_sparc/gr712rc-extra' finished successfully (4m2.628s)
ditto@RTEMS:/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems$ ./waf install
Waf: Entering directory `/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems/build'
Waf: Leaving directory `/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems/build'
'install' finished successfully (0.078s)
Waf: Entering directory `/opt/rtems/rtems-6-sparc-gr712rc-uni-6/src/rtems/build/
sparc/gr712rc-extra'
err: /bin/sh: 1: git: not found
- install /opt/rtems/rtems-6-sparc-gr712rc-uni-6/sparc-rtems6/gr712rc-extra/lib/
linkcmds (from bsps/sparc/leon3/start/linkcmds.gr712rc)
- install /opt/rtems/rtems-6-sparc-gr712rc-uni-6/sparc-rtems6/gr712rc-extra/lib/
include/bsp/console-polled.h (from bsps/include/bsp/console-polled.h)
- install /opt/rtems/rtems-6-sparc-gr712rc-uni-6/sparc-rtems6/gr712rc-extra/lib/
include/bsp/console-termios.h (from bsps/include/bsp/console-termios.h)
- install /opt/rtems/rtems-6-sparc-gr712rc-uni-6/sparc-rtems6/gr712rc-extra/lib/
include/ofw/ofw.h (from bsps/include/ofw/ofw.h)
- install /opt/rtems/rtems-6-sparc-gr712rc-uni-6/sparc-rtems6/gr712rc-extra/lib/
include/bsp/fdt.h (from bsps/include/bsp/fdt.h)
- install /opt/rtems/rtems-6-sparc-gr712rc-uni-6/sparc-rtems6/gr712rc-extra/lib/
include/bsp/gpio.h (from bsps/include/bsp/gpio.h)
- install /opt/rtems/rtems-6-sparc-gr712rc-uni-6/sparc-rtems6/gr712rc-extra/lib/
include/libchip/ata.h (from bsps/include/libchip/ata.h)
- install /opt/rtems/rtems-6-sparc-gr712rc-uni-6/sparc-rtems6/gr712rc-extra/lib/
```

5 RTEMS 빌드 활용

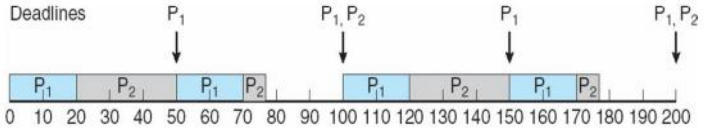
실시간성 보장을 위한 고신뢰성 커널 개발 SSLab

- **1) 스케줄링 알고리즘**
  - 기존 구현되어 있는 스케줄링 알고리즘 외,
  - 실시간 및 비-실시간 스케줄링 알고리즘을 커널 내부에 구현
  - → 선택 가능한 스케줄링 정책을 확장
  - 예)
    - **Least Laxity First (LLF)**: 작업의 남은 시간과 기한을 비교하여, 여유 시간이 적은 프로세스를 실행
    - **Weight Round Robin (WRR)**: Round Robin과 유사하지만, 각 작업에 가중치를 동적으로 할당하여 중요한 작업이 더 자주 선택되도록 조절
- **2) 스케줄링 기능 개발**
  - Dynamic Task Migration
    - 멀티코어 시스템에서 코어 간 작업을 동적으로 할당, 특정 코어에 작업 부하가 과도하게 쏠리는 현상을 방지
  - Mixed-Criticality Scheduling
    - 다양한 우선순위를 가진 작업들이 같은 시스템에서 실행될 때, 중요도가 높은 작업이 리소스를 우선적으로 사용하고, 중요도가 낮은 작업은 자원이 부족할 때 성능을 제한

산출물: 고신뢰성 실시간 커널 소스 코드

실시간성 보장을 위한 고신뢰성 커널 개발 SSLab

- **1) 스케줄링 알고리즘 지원**
  - **향후 계획**
    - 현재 RTEMS에서 지원되는 스케줄러
      - » **Priority**(우선 순위) 스케줄러
      - » **EDF**(Earliest Deadline First) 스케줄러: **Real-time**
      - » **CBS**(Constant Bandwidth Server) 스케줄러: CPU 대역폭 기반



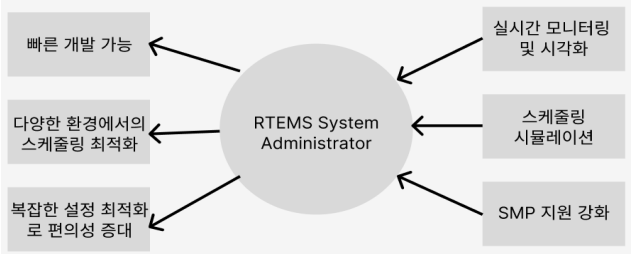
<EDF scheduling example>

- **RTEMS 스케줄러 지원 확장**
  - » **RM**(Rate Monotonic) 스케줄러 등
- **RTEMS Scheduling Framework 개발 검토**

> RTEMS provides a plugin framework
> which allows it to support multiple scheduling algorithms.

멀티태스킹 지원 및 스케줄 관리 기능 SSLab

- **1) RSA(RTEMS System Administrator) 개발 방법 및 절차**
  - 실시간 모니터링 및 시각화
    - RTEMS API 분석 및 활용, 스케줄링 상태, CPU 사용률, 작업 상태 등 모니터링
    - WebSocket 등을 통한 실시간 데이터 전송
  - 스케줄링 시뮬레이션
    - 시뮬레이션 엔진 개발, 각 스케줄링 정책의 성능 평가
    - 애플리케이션 워크로드에 맞는 스케줄링 정책 제안
  - SMP 지원 강화
    - RTEMS의 SMP 지원 수준 분석, 워크로드 및 시스템 상태 분석
    - CPU 친화도 등 최적화 알고리즘 설계, 최적 설정 제안



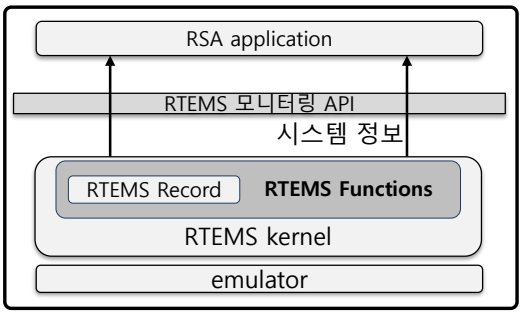
산출물: RSA 패널(S/W), 매뉴얼 및 튜토리얼(문서)

멀티태스킹 지원 및 스케줄 관리 기능 SSLab

- **1) RSA(RTEMS System Administrator) 설계 및 구현**
  - **개발 목적**
    - RTEMS 시스템 실시간 모니터링, 시각화 및 관리 기능 제공
  - **개발 배경**
    - 기존 RTEMS 관리 도구의 부족한 점 → 실시간 모니터링, 직관적인 UI 미비
    - 개발 환경 : Linux Ubuntu 기반 **RTEMS QDP** GR712RC UNI V6
  - **주요 기능**
    - **실시간 모니터링 및 시각화**
    - 스케줄링 분석 및 시뮬레이션
    - SMP 지원 강화

멀티태스킹 지원 및 스케줄 관리 기능 SSLab

- **1) RSA(RTEMS System Administrator) 설계 및 구현**
  - **실시간 모니터링 및 시각화**
    - **RSA 구조 설계**



<RSA 구조>

    - **모니터링 요소**
      - » 시스템 자원 → 시스템 자원 사용에서 예기치 못한 오류 감지
      - » 이벤트 → OS 내부 스케줄링 이벤트를 실시간으로 검증 및 확인

분류	항목
시스템 자원	CPU 사용량, 메모리 사용량, 스택 사용량, 힙 사용량
이벤트	태스크 상태 변경, 태스크 생성 및 삭제, 스케줄링 이벤트

6 참고문헌

참고문헌

SSLab

- [1] 과학기술정보통신부, 2023 우주산업실태조사, 2023.12.
- [2] Bhisale, V, Bhardwaj, K and Gavrilovska, A, "Toward LooslyCoupled Orchestration for the LEO Satellite Edge," 3rd USENIX Workshop on Hot Topics in Edge Computing (HotEdge20), 2020.
- [3] QDP GR712RC SMP Software configuration file document V6, 2020.
- [4] RTEMS User Manual, 2024
- [5]김범식, 양회석. (2021). 실시간 임베디드 시스템 환경에서 RM 스케줄러를 이용한 태스크 수준의 메모리 결함 허용 기법. 한국통신학회논문지, 46(4), 757-766. 10.7840/kics.2021.46.4.757
- [6]https://docs.rtems.org/docs/6.1/eng/