

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				

## 소프트웨어 단위시험 계획서

수정일자	수정자	버전	추가/수정 항목	내 용
2021-12-10	이종훈	0.1		초안작성
2021-12-13	이종훈	0.7		
2021-12-24	이종훈	1.0		

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				

## <제목>

I. 목적	3
II. 범위	3
III. 용어 정의 및 약어	3
IV. 참고 문서	3
V. 시험 대상	4
VI. 시험항목	4
VII. 시험 방법	4
가. test-thd-01	4
나. test-thd-02	6
다. test-thd-03	8
라. test-thd-04	9

## <표>

[표 1] 고등급 보안 마이크로 커널(CHAOS) 단위시험 항목 및 설명	4
[표 2] test-thd-03에 대한 예상 결과	8
[표 3] test-thd-04에 대한 예상 결과	9

## <그림>

[그림 1] test-thd-01에 대한 예상 결과	5
[그림 2] test-thd-02-Task T1에 대한 예상 결과	7
[그림 3] test-thd-02-Task T2에 대한 예상 결과	7
[그림 4] test-thd-02-Task T3에 대한 예상 결과	7

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				

## I. 목적

본 보고서는 고등급 보안 마이크로 커널(CHAOS)의 결함을 개발 초기에 진단하고 방지하기 위해 단위시험에 대한 계획을 기술하는 것에 목적이 있다. 이것을 통해 단위시험의 시험 대상을 정하고 구체적인 계획을 수립한다.

## II. 범위

본 보고서는 고등급 보안 마이크로 커널(CHAOS) 개발을 위해 SW 단위시험 계획의 범위를 다음과 같이 정의한다.

chsys.h(시스템 관련 매크로 및 구조)

chthreads.h(스레드 관련 매크로 및 구조)

chsched.h(스케줄러 매크로 및 구조)

chthreads.c(스레드 코드)

## III. 용어 정의 및 약어

없음

## IV. 참고 문서

- IEC, IEC61508. “61508 functional safety of electrical/electronic/programmable electronic safety-related systems.” International electrotechnical commission (1998).
- 소프트웨어 안전 기능 요구사항 명세서 (2019)
- 소프트웨어 안전 기능 설계명세서 (2020)
- SW 안전성 공통 개발 가이드. “정보통신산업진흥원.” (2016).
- ChibiOS/RT The Ultimate Guide (RT 6)  
<https://www.chibios.org/dokuwiki/doku.php?id=chibios:documentation:books:rt:start>

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				

## V. 시험 대상

고등급 보안 마이크로 커널(CHAOS) 개발을 위한 단위시험 계획의 시험 대상을 스레드 코드인 chthreads.c 파일로 정하고 스레드와 관련된 안전성 기능을 다음과 같이 분류한다.

- 스레드 슬립 상태
- 스레드 우선순위 변경 논리

## VI. 시험항목

고등급 보안 마이크로 커널(CHAOS) 개발을 위해 SW 단위시험 계획의 시험 항목을 다음 표 1과 같이 정의한다.

ID	시험 항목	설명
test-thd-01	Thread Sleep functionality	chThdSleep() 함수의 기능이 정상적으로 작동하는지 확인한다.
test-thd-02	Ready List functionality	여러 스레드를 실행시킨 후 올바른 우선순위 순서로 작동하는지 확인한다.
test-thd-03	Priority Change test	스레드의 우선순위를 변경했을 경우 올바르게 변경되는지 확인한다.
test-thd-04	Priority Change test with Priority Inheritance	우선순위 상속에 의해 스레드의 우선순위를 변경했을 경우 올바르게 변경되는지 확인한다.

[표 1] 고등급 보안 마이크로 커널(CHAOS) 단위시험 항목 및 설명

## VII. 시험 방법

### 가. test-thd-01

#### 1) 입력 데이터

가) Task T1

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				

- 100 system tick

나) Task T2

- 100000 microseconds

다) Task T3

- 100 milliseconds

라) Task T4

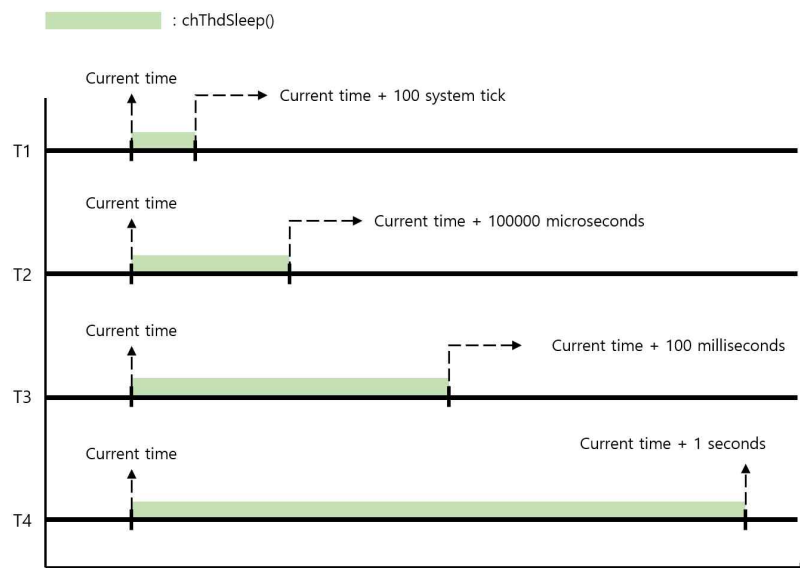
- 1 second

## 2) 시험 방법

현재 시스템 시간을 확인한 후 스레드를 100 system tick, 100000 microseconds, 100 milliseconds, 1 seconds 씩 chThdSleep() 함수를 사용하여 슬립 상태를 지속시키고 종료 시에 시스템 시간을 다시 확인하여 할당한 시간만큼 슬립 상태가 유지되는지 확인한다.

## 3) 예상 결과

그림 1은 test-thd-01에 대한 동작과 예상 결과를 표로 도식화한 그림이고 각 스레드는 현재 시간을 측정하고 chThdSleep() 함수를 호출하여 할당한 시간만큼 슬립 상태를 유지한다. sleep 수행 후에 시간을 측정하면 할당한 시간만큼 슬립 상태를 유지할 것으로 예상된다.



[그림 1] test-thd-01에 대한 예상 결과

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				

## 나. test-thd-02

### 1) 입력 데이터

#### 가) Task T1

- thread[0] : prio - 4, 출력값 = “D”
- thread[1] : prio - 3, 출력값 = “C”
- thread[2] : prio - 2, 출력값 = “B”
- thread[3] : prio - 1, 출력값 = “A”

#### 나) Task T2

- thread[3] : prio - 1, 출력값 = “A”
- thread[2] : prio - 2, 출력값 = “B”
- thread[1] : prio - 3, 출력값 = “C”
- thread[0] : prio - 4, 출력값 = “D”

#### 다) Task T3

- thread[1] : prio - 1, 출력값 = “C”
- thread[0] : prio - 2, 출력값 = “D”
- thread[3] : prio - 3, 출력값 = “A”
- thread[2] : prio - 4, 출력값 = “B”

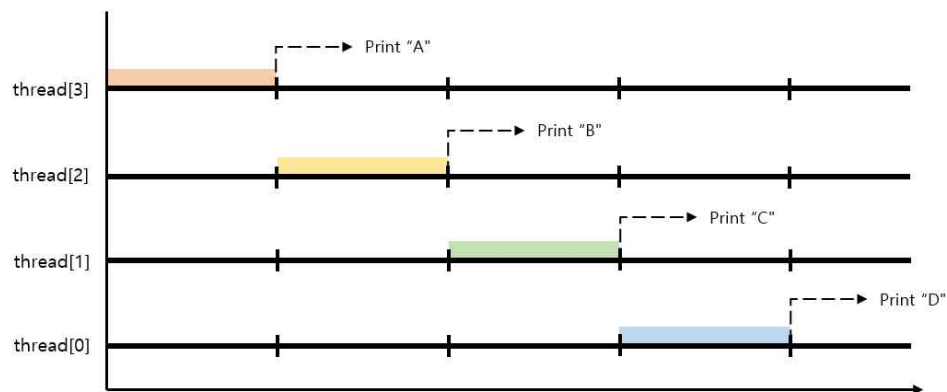
### 2) 시험 방법

Task 1은 우선순위를 높은 순서대로, Task 2는 우선순위가 낮은 순서대로, Task 3는 무작위로 4개의 스레드를 만들어 실행 순서를 출력값으로 확인한다.

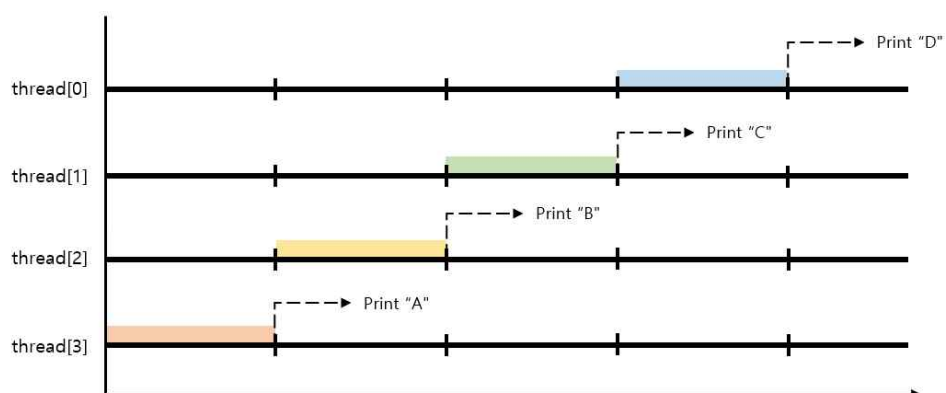
### 3) 예상 결과

그림 2, 그림3, 그림4는 test-thd-02에 대한 동작 과정과 예상 결과를 표로 도식화한 그림이고 각 스레드들은 생성된 순서에 관계없이 우선순위에 따라 작업을 수행할 것으로 예상된다. 그렇기 때문에 Task T1와 Task T2, Task T3 모두 출력 결과로 ‘ABCD’가 될 것이다.

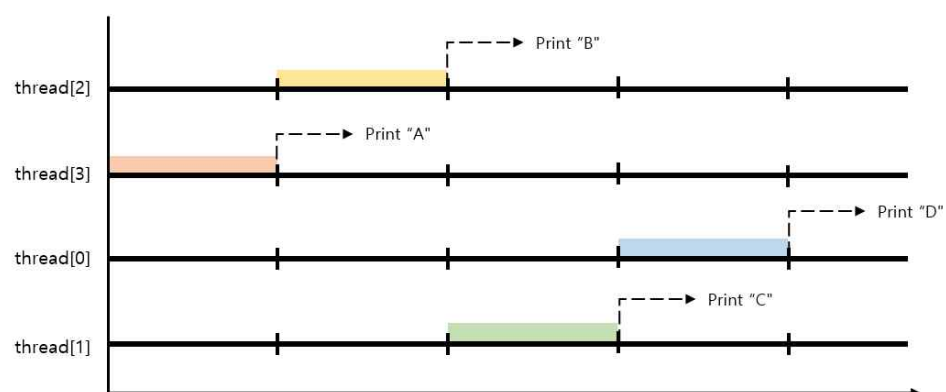
	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				



[그림 2] test-thd-02-Task T1에 대한 예상 결과



[그림 3] test-thd-02-Task T2에 대한 예상 결과



[그림 4] test-thd-02-Task T3에 대한 예상 결과

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				

## 다. test-thd-03

### 1. 입력 데이터

가) Task T1

- prio = chThdGetPriority() : 현재 작업 중인 스레드의 우선순위이다.

나) 스레드의 두 가지 우선순위 정보

- 스레드는 실제 우선순위 (Real prio)와 현재 우선순위 (Current prio)를 가지고 있으며, 기본적으로 두 가지 우선순위는 동일한 값을 가지고 있다. 우선순위 상속 프로토콜이 동작하는 상황에서 스레드의 우선순위가 부스트 될 수 있는데, 이때 부스트 되는 우선순위가 현재 우선순위 (Current prio)와 실제 우선순위 (Real prio)가 같거나 현재 우선순위(Current prio)가 부스트 되는 우선순위보다 작다면 현재 우선순위 (Current prio)와 실제 우선순위(Real prio)가 변경된다. 만약 부스트 되는 우선순위가 현재 우선순위 (Current prio)보다는 작다면 실제 우선순위 (Real prio)만 변경된다.

### 2. 시험 방법

현재 작업 중인 스레드에 현재 우선순위보다 1만큼 증가한 우선순위를 설정하고, 우선순위가 올바르게 할당되었는지 확인한다. 이 때 chThdGetPriority() 함수를 사용하여 현재 작업 중인 스레드의 우선순위를 받아 저장하고 새로운 우선순위를 설정할 때 chThdSetPriority() 함수에 새로운 우선순위 값을 인자로 전달하여 함수를 호출한다. chThdSetPriority() 함수의 반환 값은 이전 우선순위이기 때문에 해당 반환값을 확인하고 다시 chThdGetPriority() 함수를 호출하여 새로운 우선순위가 올바르게 설정되었는지 확인한다. 그리고 우선순위를 이전 값으로 재설정 후 우선순위가 올바르게 되돌아왔는지 확인하기 위해서 chThdSetPriority() 함수에 이전 우선순위를 입력하고 chThdGetPriority() 함수를 호출하여 원래 우선순위와 동일한지 확인한다.

### 3. 예상 결과

표 2는 test-thd-03의 예상 결과를 표로 도식화한 것이다.

Step1은 현재 우선순위보다 높은 우선순위를 설정하는 단계이고 Step2는 원래 우선순위로 재설정하는 단계이다. chThdGetPriority() 함수의 반환값은 cp라고 할 때 Step1과 Step2의 예상 결과는 표 2와 같이 정상적으로 동작할 것이다.



	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				

Step	prio	Current Prio	Real Prio	Old Prio
1	cp	cp	cp	
	cp	cp+1	cp	cp
	cp	cp+1	cp	cp
	cp	cp+1	cp	cp
2	cp	cp	cp+1	cp+1
	cp	cp	cp+1	cp+1
	cp	cp	cp+1	cp+1

[표 2] test-thd-03에 대한 예상 결과

## 라. test-thd-04

### 1. 입력 데이터

다) Task T1

- prio = chThdGetPriority() : 현재 작업 중인 스레드의 우선순위이다.

### 2. 시험 방법

먼저 현재 우선순위가 실제 우선순위보다 큰 상황을 만들고 chThdGetPriority() 함수를 통해 확인한다. 그 다음 chThdSetPriority() 함수를 사용하여 현재 우선순위를 원래보다는 낮고 실제 우선순위보다는 높게 설정한다. 이후 이전 우선순위가 prio와 같은지 확인하고 실제 우선순위만 설정한 값과 같게 변경되었는지 확인한다. 다음 단계로는 다시 chThdSetPriority() 함수를 사용하여 현재 우선순위를 원래 우선순위와 실제 우선순위보다 높게 설정한다. 이후 이전 우선순위가 prio+1과 같은지 확인하고 현재 우선순위, 실제 우선순위가 둘 다 설정한 값과 같게 변경되었는지 확인한다.

### 3. 예상 결과

표 3은 test-thd-04의 예상 결과를 표로 도식화한 것이다. Step1은 현재 우선순위가 실제 우선순위보다 큰 상황을 만드는 단계이고 Step2는 현재 우선순위를 원래 우선순위보다는 낮고 실제 우선순위보다는 높게 설정한 뒤 확인하는 단계이다. Step3는 현재 우선순위를 원래 우선순위와 실제 우선순위보다 높게 설정한 뒤 확인하는 단계이다.

	고등급(EAL6 이상) 보안마이크로커널 개발					
	작성자	이종훈	소속	고려대학교	연구 책임자	김승주
	작성일	2021-12-22	파일명	2021-기술문서03-SW단위시험계획서-v2.hwp		
	제목	소프트웨어 단위시험 계획서				

Step	prio	Current Prio	Real Prio	Old Prio
1	cp	cp	cp	
	cp	cp+2	cp	
	cp	cp+2	cp	
2	cp	cp+2	cp+1	cp
	cp	cp+2	cp+1	cp
	cp	cp+2	cp+1	cp
	cp	cp+2	cp+1	cp
3	cp	cp+3	cp+3	cp+1
	cp	cp+3	cp+3	cp+1
	cp	cp+3	cp+3	cp+1
	cp	cp+3	cp+3	cp+1

[표 3] test-thd-04에 대한 예상 결과