

모바일 환경에서 하이퍼바이저 기반의 안드로이드 커널 개발

2012. 12. 14

(주)알투소프트

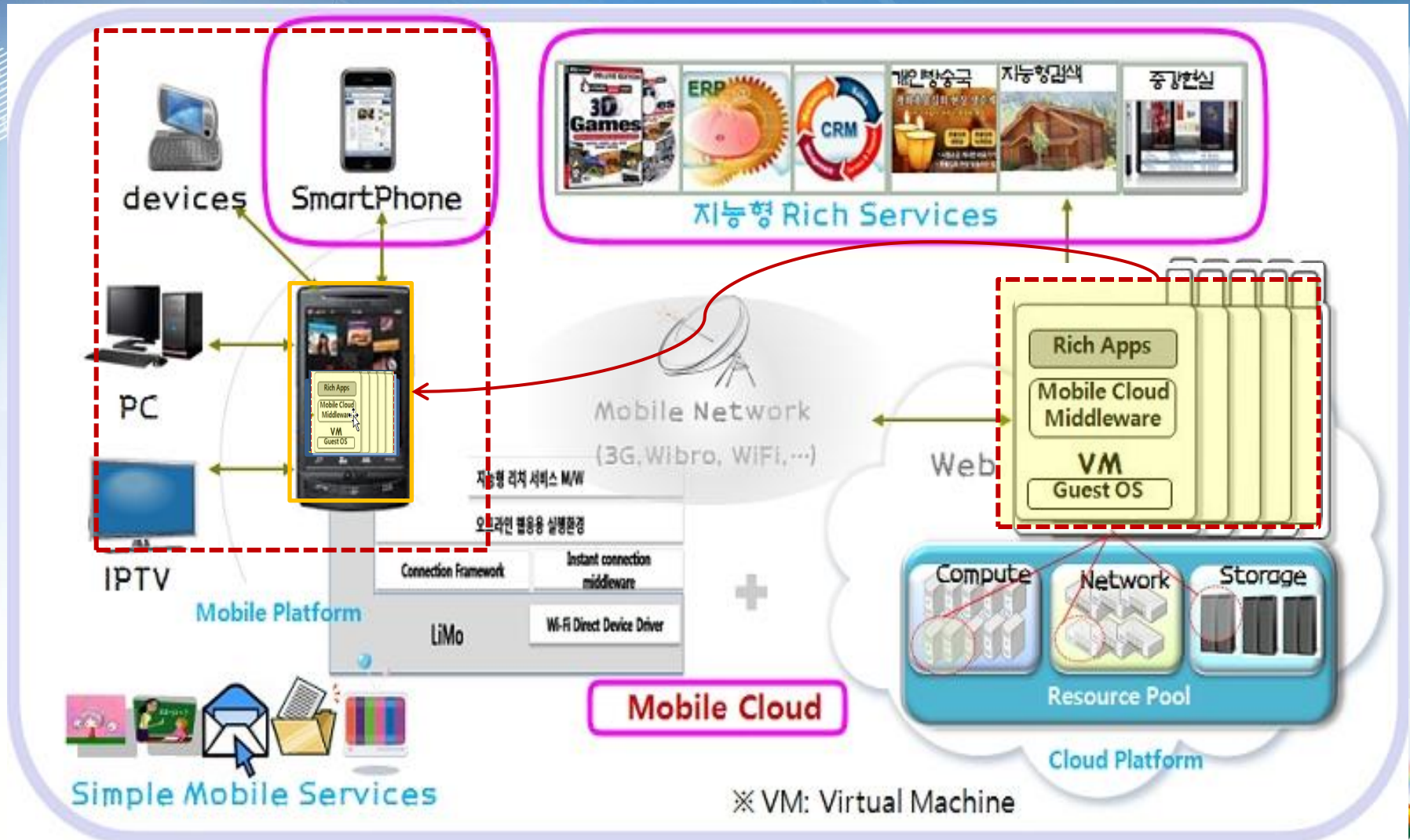
목 차

- Ⅰ 과제 개요
- Ⅱ 개발 목표 및 내용
- Ⅲ 과제 수행 내용
- Ⅳ 최종 결과 및 상용화
- Ⅴ 사업비



1.1 개발과제의 개요

1. 과제 개요



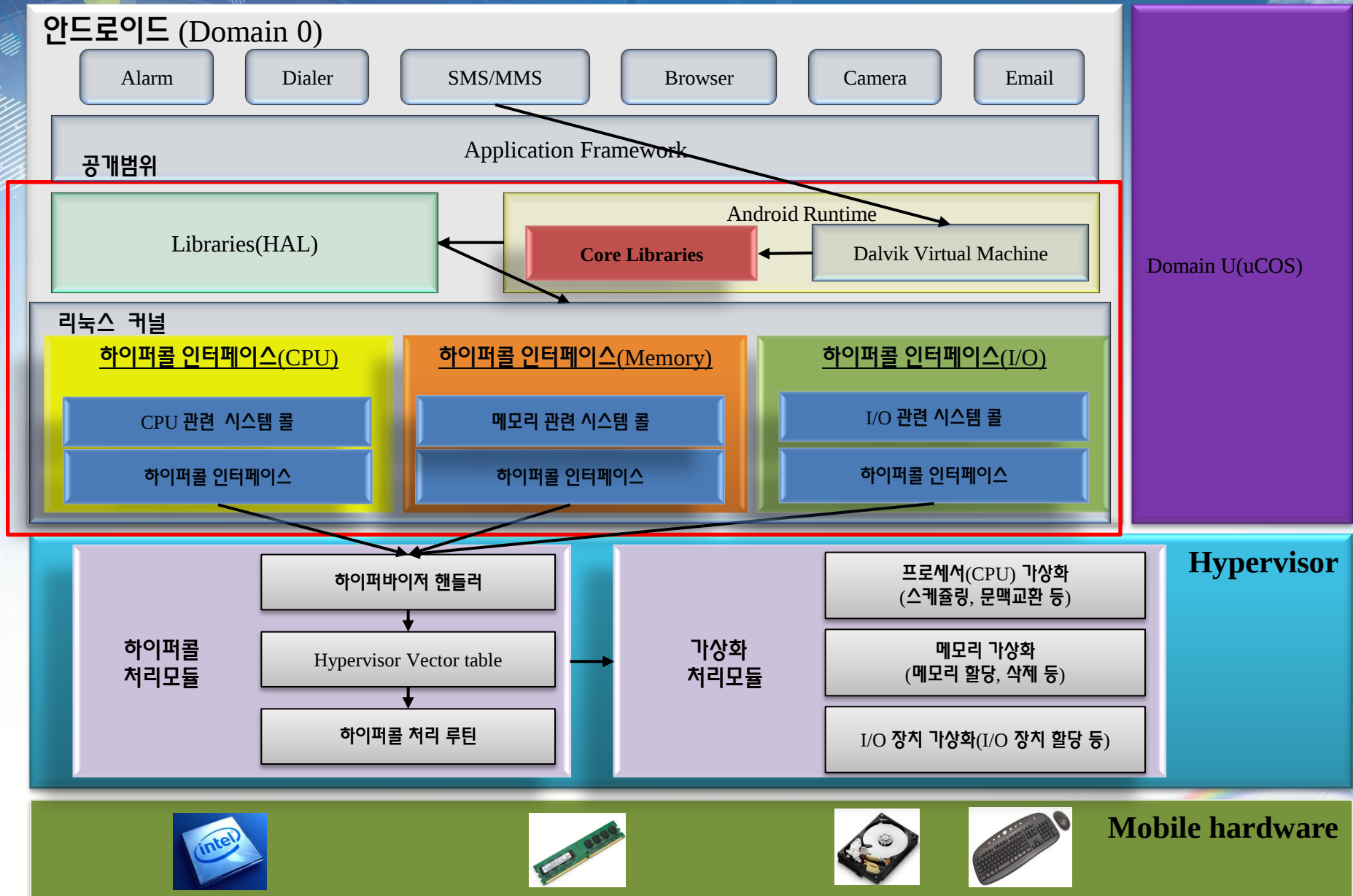
모바일 환경에서 하이퍼바이저 기반의 안드로이드 커널 개발

- 프로세스(CPU) 가상화를 위한 하이퍼콜 인터페이스 개발
- 메모리 가상화를 위한 하이퍼콜 인터페이스 개발
- I/O 장치 가상화를 위한 하이퍼콜 인터페이스 개발



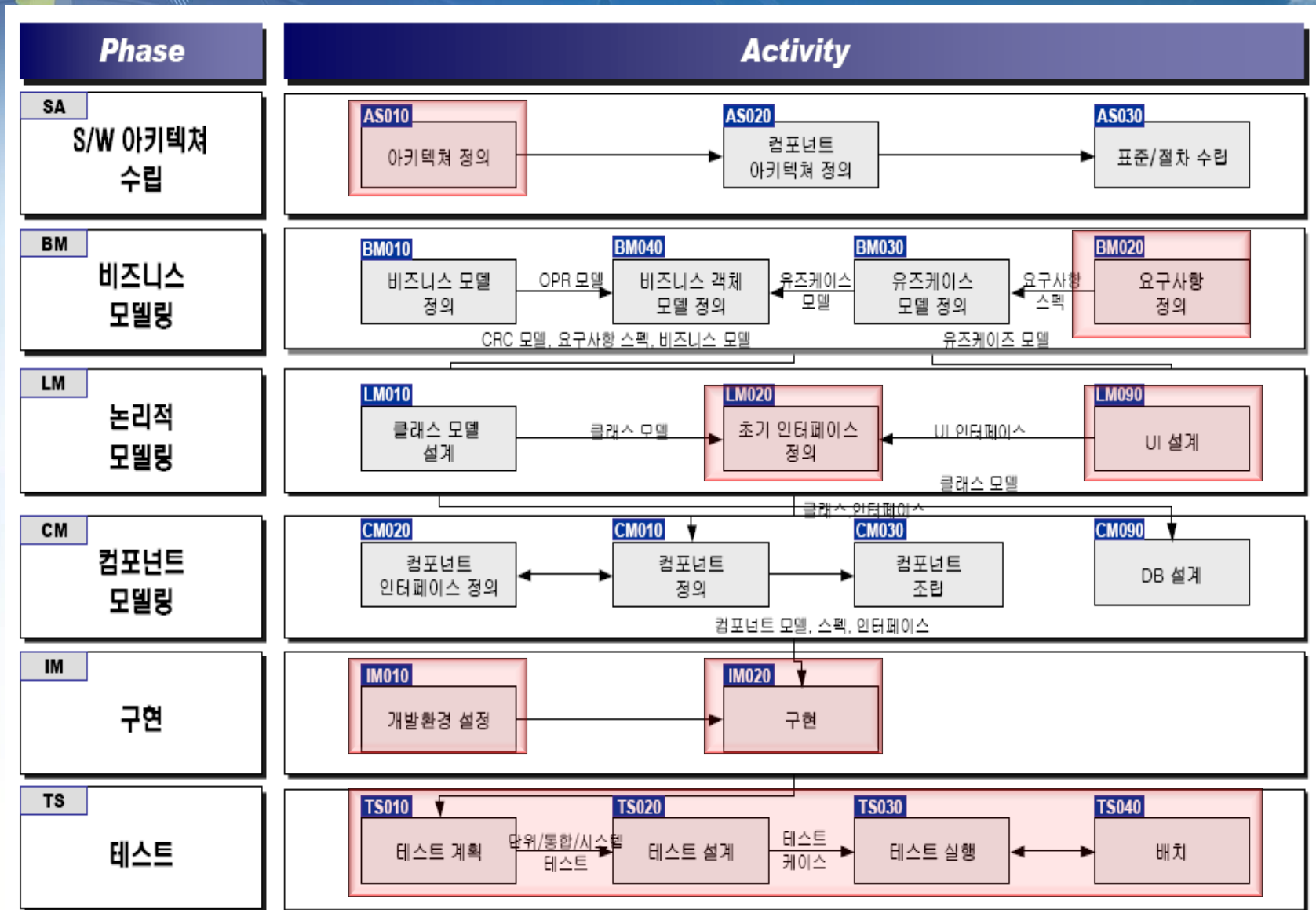
2.2 시스템 구성

2. 개발목표 및 내용

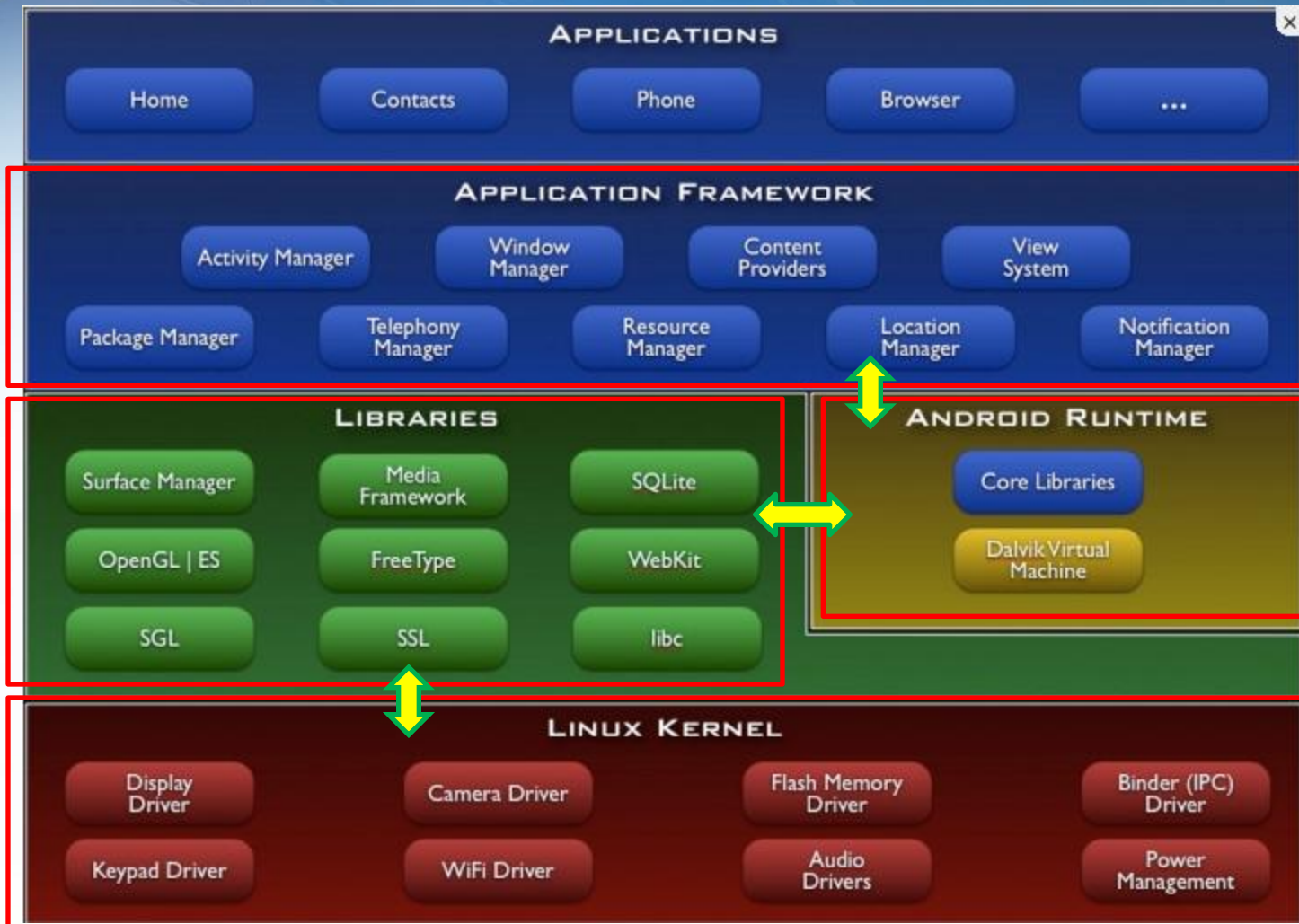


3.1 연구 개발 방법론

3. 파제 수행 내용



3.2 안드로이드 분석



3.3 안드로이드와 하이퍼바이저의 연계 방안

3. 과제 수행 내용

고려 사항

- 운영체제의 기본적 기능과 하이퍼바이저와의 연계
- Bionic 라이브러리와 하이퍼바이저 라이브러리간의 호환성
- Dalvik VM에서 발생하는 시스템 콜에 대한 처리 방안

해결 방안

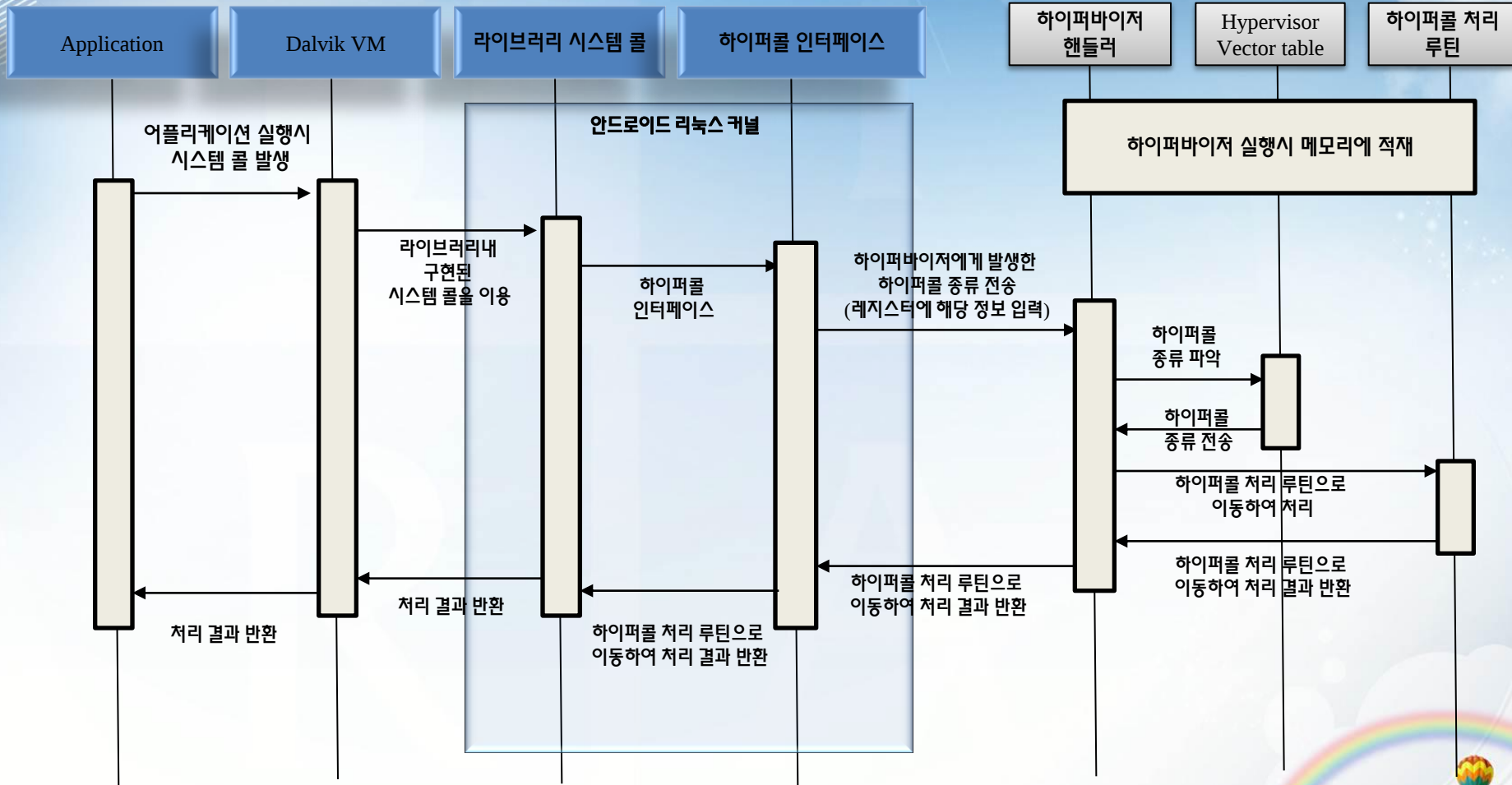
- 운영체제의 기본적 기능
 - 리눅스 커널의 시스템 콜 분석 및 하이퍼바이저와 연계
- Bionic 라이브러리와 하이퍼바이저 라이브러리간의 호환성
 - Bionic 라이브러리 내에 하이퍼바이저의 하이퍼콜 인터페이스 호출 구현
 - 하이퍼바이저의 라이브러리에 Bionic 라이브러리 구현
- Dalvik VM에서 발생하는 시스템 콜에 대한 처리 방안
 - Dalvik VM 내의 코어 라이브러리(JNI)를 통해 IPC 시스템 콜 명령이 수행하도록 변경

3.4 하이퍼바이저와 안드로이드의 하이퍼콜 처리 흐름

3. 과제 수행 내용

안드로이드

하이퍼바이저



3.5 하이퍼콜 인터페이스 매핑 테이블

3. 과제 수행 내용

주요 기능		안드로이드 시스템 콜	하이퍼바이저 하이퍼콜
CPU 가상화	쓰레드(프로세스) 처리	fork(), exec(), pthreads()	sys_thread_create sys_thread_join sys_thread_exit
	인터럽트 처리 기능	interrupt()	sys_interrupt_register sys_interrupt_deregister sys_interrupt_wait
	상호배제 기능	futex_signal(), futex_wait()	sys_futex_wait sys_futex_signal
	프로세스간 통신	ipc()	sys_ipc_send sys_ipc_recv
메모리 가상화	메모리 페이지 접근 기능	mmap()	sys_memsections
I/O 가상화	I/O 처리 핸들러	interrupt()	sys_vio_handler
	실제 장치 드라이버 구현	/dev/input	/driver
Dalvik VM 내 IPC 처리	IPC 처리 기능	ipc()	sys_ipc_send sys_ipc_recv



3.6 가상화 하이퍼콜 인터페이스 구현

기능	내용
fork() 하이퍼콜 인터페이스	<pre> .text .type __fork, #function /* if system call is fork .globl __fork /* fork execution routine jump .align 4 .fnstart __fork: /* fork execution => modified source(hypercall interface)*/ ldr r0, =__fork_text /* r0 register <- vector value input */ bl hypervisor_handler /* move to hypervisor_handler in hyperviosr */ b _Exit .fnend </pre>
하이퍼바이저 핸들러 호출을 위한 벡터 테이블	<pre> .section .vectors, "aw" .global vectors vectors: b __kernel_main /* "Reset" pin set high on processor. */ b unhandled_vector /* Undefined instruction. */ b hypervisor_handler /* syscall. */ b unhandled_vector /* Prefetch abort exception. */ ... </pre>



3.6 가상화 하이퍼콜 인터페이스 구현

3. 과제 수행 내용

기능	내용
하이퍼바이저 핸들러	<pre> noretun_function hypervisor_handler ... /* Convert system call number into a jump vector. */ adr lr, hypercall_vector and r12, r12, #(NUM_SYSCALLS - 1) add r12, lr, r12, LSL #2 /* Load the call. */ ldr r12, [r12] /* Jump to the syscall. */ ... end_function hypervisor_handler </pre>
하이퍼바이저 벡터 테이블	<pre> list hypercall_vector .word sys_thread_myself /* 0000 */ .word sys_thread_create /* 0001 */ .word sys_thread_exit /* 0002 */ .word sys_futex_wait /* 0003 */ .word sys_futex_signal /* 0004 */ ... </pre>



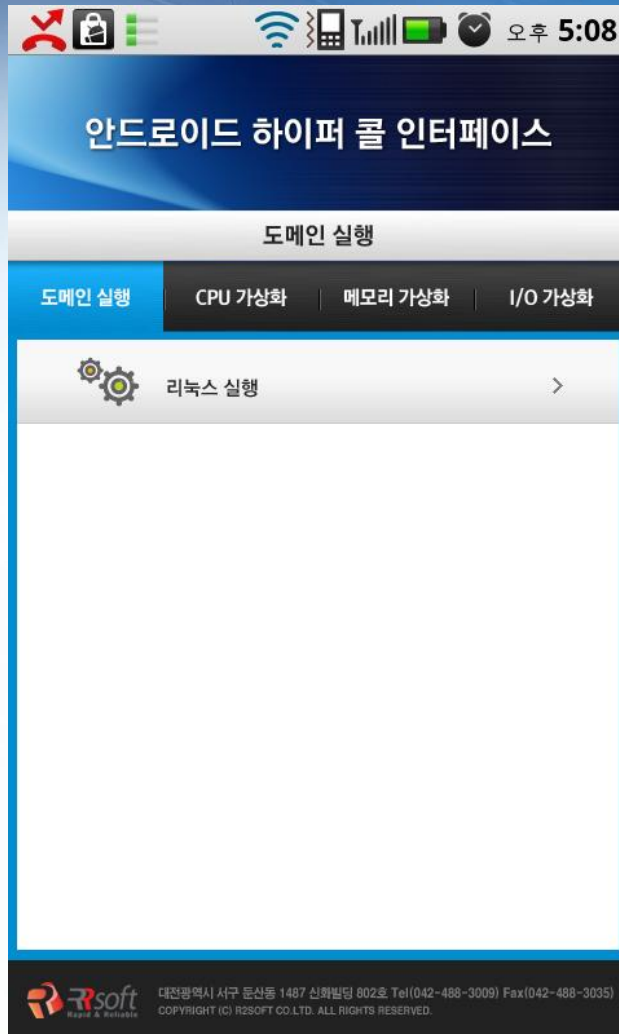
3.6 가상화 하이퍼콜 인터페이스 구현

기능	내용
하이퍼바이저의 하이퍼콜 인터페이스 처리 루틴	<pre> sys_thread_create(word_t pc, word_t sp, word_t r0, word_t prio) { /* Is the priority correct? */ if (EXPECT_FALSE(prio > MAX_PRIORITY)) syscall_return_error(-1, EINVAL); spin_lock(&tcb_lock); /* Are there any free TCBs present? */ if (EXPECT_FALSE(tcb_free_head == NULL)) { spin_unlock(&tcb_lock); syscall_return_error(-1, ENOMEM); } /* Dequeue the first TCB in the list. */ pre_create_thread_callbacks(new_tcb); tcb_t *next = activate_schedule(new_tcb); post_create_thread_callbacks(new_tcb); spin_unlock(&tcb_lock); /* Switch to the highest thread, returning the new thread's tid. */ set_syscall_return_val_success(current_tcb, new_tcb->tid); switch_to(next); } </pre>



3.7 가상화 하이퍼콜 사용자 인터페이스

3. 과제 수행 내용



3.7 가상화 하이퍼콜 사용자 인터페이스

3. 과제 수행 내용

안드로이드 하이퍼 콜 인터페이스

메모리 가상화

도메인 실행 | CPU 가상화 | **메모리 가상화** | I/O 가상화

메모리 페이지 테이블 요청

- 메모리 페이지 테이블 요청 시작
- 메모리 페이지 테이블
- 메모리 페이지 테이블 요청 종료

메모리 페이지 테이블 정보 부

도메인명	메모리테이블 주소	할당량	할당율
안드로이드	0X48000000	512MB	30%
리눅스	0X48004000	512MB	30%

Rsoft Rapid & Reliable
대전광역시 서구 둔산동 1487 신화빌딩 802호 Tel(042-488-3009) Fax(042-488-3035)
COPYRIGHT (C) R2SOFT CO.LTD. ALL RIGHTS RESERVED.

안드로이드 하이퍼 콜 인터페이스

I/O 가상화

도메인 실행 | CPU 가상화 | 메모리 가상화 | **I/O 가상화**

물리 장치 가상화 요청

- 물리 장치 가상화 시작
- I/O 장치(IRQ 처리 및 물리장치 할당)
- 물리 장치 가상화 종료

가상화 물리 장치 목록

도메인명	소유 장치명	가상화 여부	가상화 요청
안드로이드	모니터	Y	
안드로이드	네트워크	Y	
안드로이드	디스크	Y	
리눅스	모니터	N	
리눅스	네트워크	N	
리눅스	디스크	N	

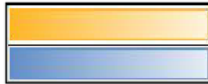
Rsoft Rapid & Reliable
대전광역시 서구 둔산동 1487 신화빌딩 802호 Tel(042-488-3009) Fax(042-488-3035)
COPYRIGHT (C) R2SOFT CO.LTD. ALL RIGHTS RESERVED.

3.8 테스트 결과

테스트 결과

- 전문 품질 검증 회사에 테스트 의뢰(2012년 11월 28일 ~ 30일)
- I/O 가상화 부분에 일부 기능이 미구현됨으로써 기능성 71%, 비기능성 78%의 평가결과를 획득

평가 결과



최초 테스트 결과
최종 테스트 결과

구분	품질 특성별 평가기준	최종 평가 기준
기능성	매우미흡: 50%미만, 미흡: 90%미만, 양호: 100% 미만, 탁월: 100%	전체 품질 특성별 평가 결과 값 중 가장 낮은 평가 결과를 최종 평가 등급으로 정함 (평가 등급 : 탁월, 양호, 미흡, 매우 미흡)
비기능성	매우미흡: 50%미만, 미흡: 85%미만, 양호: 95% 미만, 탁월: 95% 이상	

구분	품질 특성	평가 결과	평가 값	평가 기준(적합 기준)	측정 방법
기능성	기능 적합성	미흡	71%	71% (적합 평가 기준: 90%)	- 기능 구현 커버리지 - 요구 사양(사업계획서)에 제시한 기능의 구현을 완료하였는가를 확인 - 기능 타당성 - 특정 작업을 수행하기 위해 구현한 기능이 문제 없이 수행되는가를 확인
		미흡	71%	71%	
비 기능성	사용성	매우미흡	22%	22%	학습성 - 기능이 사용자 설명서 또는 도움말 기능에 올바르게 설명되어 있는 비율
		미흡	78%	78%	
	신뢰성	매우미흡	0%	0%	장애 제거 - 발견된 결함이 수정된 비율

(적합 평가 기준: 85%)

최종 평가	☐ 탁월	- 안드로이드 커널 개발에서 3가지 인터페이스 중 CPU, 메모리 인터페이스는 개발되었으나 I/O 인터페이스는 기능이 미구현 상태임 - I/O 인터페이스에서 I/O 이벤트 처리 기능은 50%, 게스트 운영체제의 물리적 장치 접근 기능은 10% 개발되었음 - 가상화 커널로 구현된 Android 운영체제에서 다양한 App들은 정상적으로 실행되었음 - 요구사항 명세서 및 사용자 매뉴얼에 CPU 가상화 하이퍼콜 인터페이스 내용이 누락되어 있음
	☐ 양호	
	☒ 미흡	
	☐ 매우미흡	



4.1 과제 최종 결과

최종 개발 결과물

- ❖ CPU 가상화 하이퍼콜 인터페이스 및 메모리 가상화 하이퍼콜 인터페이스 구현은 성공적으로 완료(개발 결과 소스)
 - I/O 가상화 하이퍼콜 인터페이스 부분에서 미구현 기능 일부 존재
 - I/O 장치 가상화를 위한 하이퍼바이저 내 장치 드라이버 개발(20%)
 - I/O 장치 가상화를 위한 하이퍼콜 인터페이스 구현(50%)
 - I/O 장치 가상화를 위한 인터럽트 처리 루틴 구현(100%)
- ❖ 특허 1건 출원(공개특허)
- ❖ 공개 SW 포탈(www.sourceforge.net)에 공개 SW 프로젝트 등록 및 운영 중
- ❖ 개발 결과 산출물
 - ❖ 설계 문서 및 품질테스트 결과서



4.2 기대효과

기대효과

- ❖ 개발 기술의 공개를 통해 모바일 가상화 시장 활성화 및 관련 연구의 중복 개발 방지
- ❖ 모바일 환경뿐만 아니라 임베디드 환경(차량, 가전 등)에서 안드로이드를 사용하려는 분야에 활용 가능
 - ❖ 모바일 환경에서의 보안 솔루션, 차량용 안드로이드 앱 개발, 차량 제어 시스템 개발 등
- ❖ 하이퍼바이저는 일부 공개 SW로 제공하고 있으나 안드로이드와의 연계방안은 공개되지 않아 관련 개발 인력들의 기술력 향상이 미흡했던 문제 해결



4.3 상용화

상용화 계획

❖ 개발 결과물 활용

- 미구현 기능 개발 완료
 - 미구현 부분 개발 완료 및 자체 테스트(2012년 12월 31일)
- 오류 분석에 의한 보완 작업(2013년 1월)
 - 오픈소스 포털(www.sourceforge.net)에 version 1.0 등록
 - 안드로이드 버전 변경에 따른 지속적인 하이퍼콜 인터페이스 업데이트(2013년 1월 이후)

❖ 시험 인증 획득(2013년 1월 ~ 3월)

- 신뢰성 확보
 - 한국정보통신기술협회(TTA) SW 시험 인증
 - 공개 SW 역량프라자 및 전문 컨설팅 기업/기관과 연계한 SW 프로세스 품질관리, 테스트, 라이선스 검증

❖ 모바일 가상화 환경의 안드로이드 보안 플랫폼 개발 (2013년 4월 ~ 12월)

- 모바일 가상화 환경의 보안 플랫폼 개발 시 기반 기술로 이용

5.1 사업비 사용 실적

5. 사업비

연구 개발비

(단위:천원)

구 분 비 목	협약예산			변경예산			집행금액			잔 액			집행율 (%)	증감 사유
	현금	현물	계	현금	현물	계	현금	현물	계	현금	현물	계		
1. 인건비	108,000	54,000	162,000	108,000	54,000	162,000	108,000	54,000	162,000				100	
2. 직 접 비	65,870		65,870	65,870		65,870	65,861		65,861	9		9	99.99	재료비감소 연구활동비 증가
- 연구시설·장비 및 재료비	32,000		32,000	32,000		32,000	30,000		30,000	2,000		2,000	93.75	
- 연구활동비	33,870		33,870	33,870		33,870	35,861		35,861	-1,991		-1,991	105.88	
3. 간 접 비	2,130		2,130	2,130		2,130	1,818		1,818	312		312	85.36	
합 계	176,000	54,000	230,000	176,000	54,000	230,000	175,679	54,000	229,679	321		321	99.86	



감사합니다