Embedded System의 펌웨어 보안

4nsys 송 민 호

CodeEngn

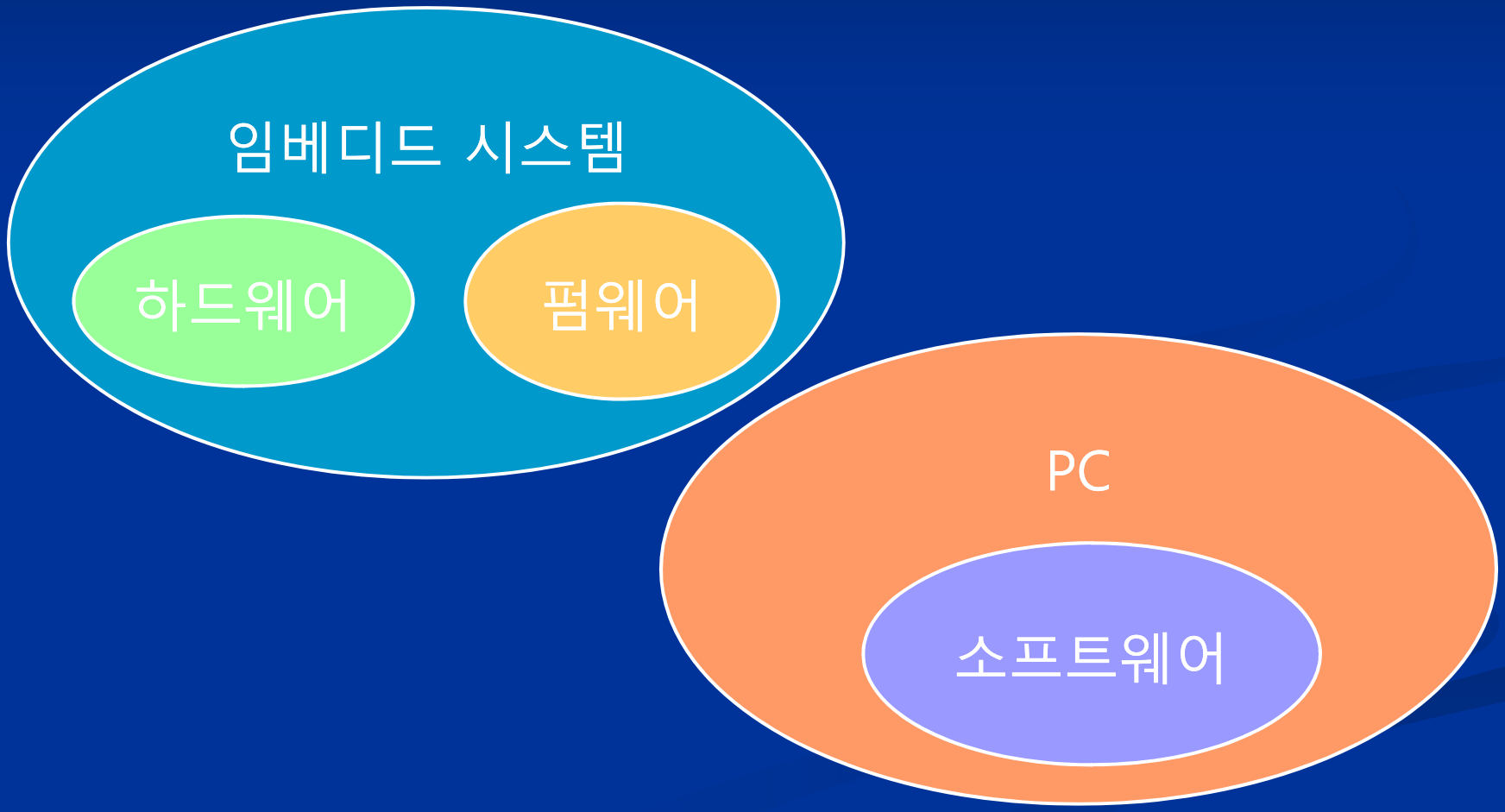
www.CodeEngn.com

목차

- 임베디드 시스템에서의 소프트웨어
- 임베디드 시스템의 메모리 동작
- MMU와 소프트웨어
- MMU에 의한 Paging의 이해
- Buffer Overflow
- 펌웨어 보호의 다른 위험요소
- 리버스 엔지니어링과 임베디드

임베디드 시스템에서의 소프트웨어

임베디드 제품의 구성



펌웨어의 운영특징

항 목	내 용	비 고
OS (Operating System)	Linux , Windows CE , Nucleous, pSOS, VxWorks , Embedded XP,	Linux는 비용이 저렴 Embedded XP 는 PC와 호환되는 환경에서만 사용가능 OS-어플리케이션 일체형도 존재
저장 미디어(Media)	Flash Memory (NAND , NOR) ROM HDD	Flash Memory의 경우 NOR + NAND 복합형도 있음
미디어 크기	일반적으로 2 ~ 16 MByte	일부 매우 큰 타입도 있음
CPU 코어	ARM , MIPS , POWER , x86계열	ARM의 경우 Thumb 모드 존재

MMU와 소프트웨어

MMU와 프로세스의 관계

페이징 미 지원 시스템

페이징 지원 시스템

Paging 사용의 특징

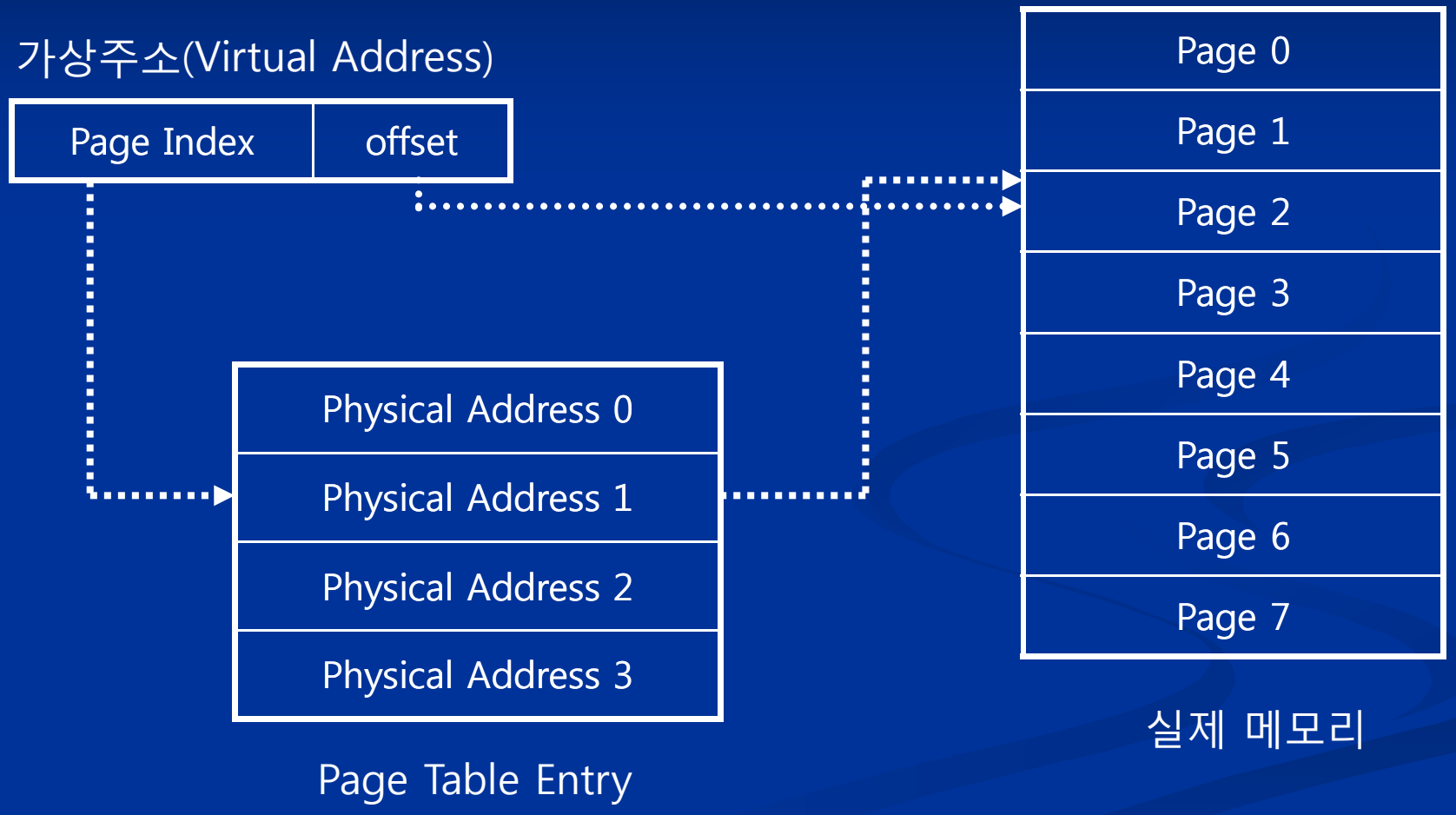
항 목	Paging	Non Paging
프로세스	독립된 여러 개의 프로세스를 만들 수 있다.	프로세스간 독립성구현이 힘들다
메모리 효율	프로세스 생성및 사용되는 메모리 할당/헤제에 의한 메모리 단편화를 최소화 할 수 있다. 페이지정보를 보관하기위한 영역으로 메모리가 일부 소비된다.	프로세스 생성/해제간 메모리단편화 발생가능성이 높다. 페이지관리를 위한 메모리 소비가 없다.
속 도	메모리를 참조하는 명령은 Page Table Entry를 검색하는 부하가 걸릴 수 있다.	영향 없다.
가상메모리	Exception-Fault를 이용한 가상메모리 구현	exception이 발생한다는 것은 메모리가 부족하다는 뜻이다.

MMU에 의한 Paging의 이해

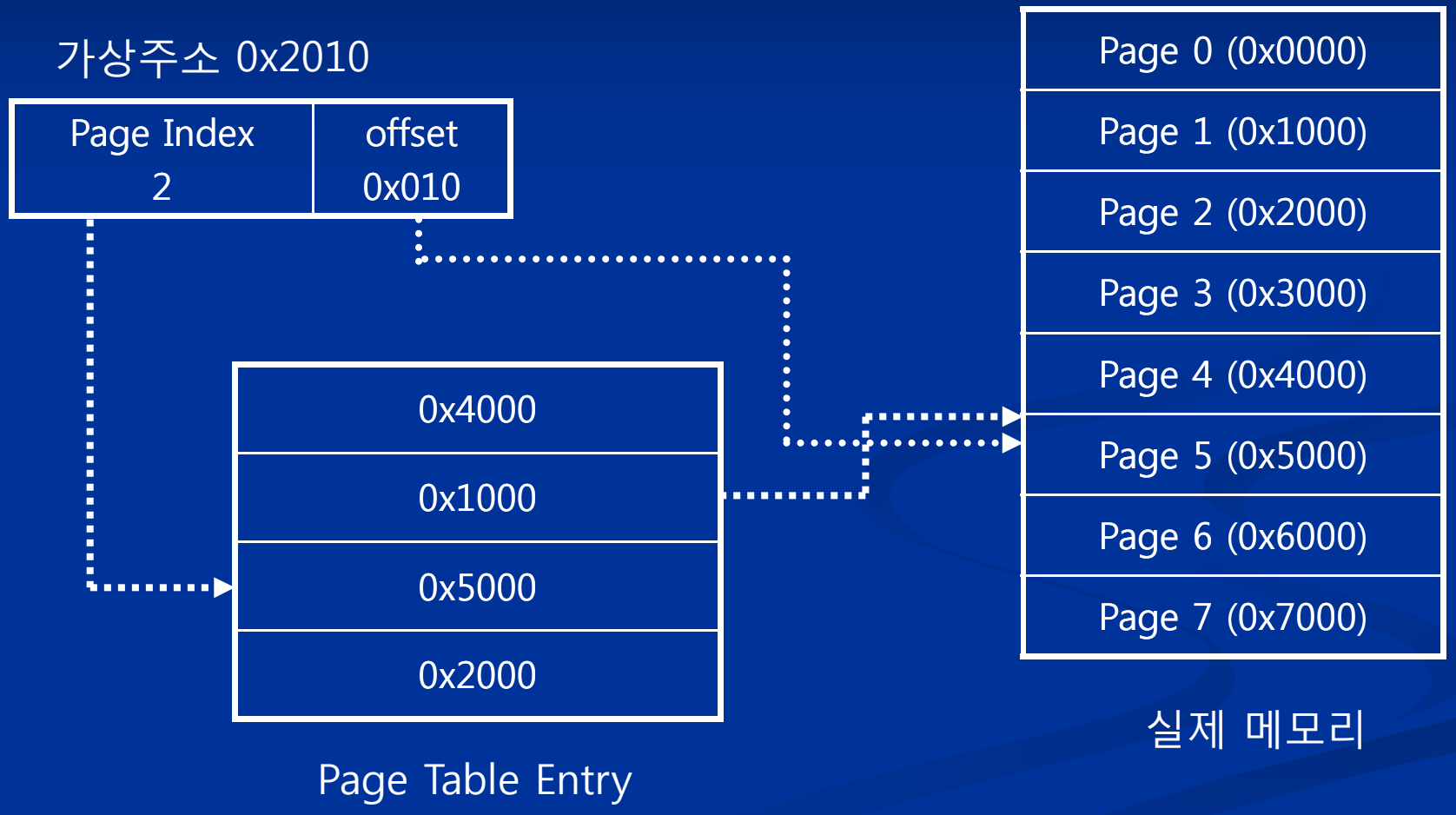
Paging 형태


빈공간
빈공간
빈공간
빈공간
빈공간
빈공간
프로세스1 (2)
프로세스1 (1)
커널(2)
커널(1)
프로세스1
빈공간
빈공간
빈공간
빈공간
빈공간
빈공간
빈공간
프로세스2 (1)
커널(2)
커널(1)
프로세스2
빈공간
빈공간
빈공간
빈공간
빈공간
프로세스3 (3)
프로세스3 (2)
프로세스3 (1)
커널(2)
커널(1)
프로세스3
프로세스 공간
빈공간
프로세스1 (2)
프로세스3 (2)
프로세스3 (3)
프로세스1 (1)
프로세스3 (1)
커널(2)
프로세스2 (1)
커널(1)
페이지앤트리
메모리

Paging 의 동작

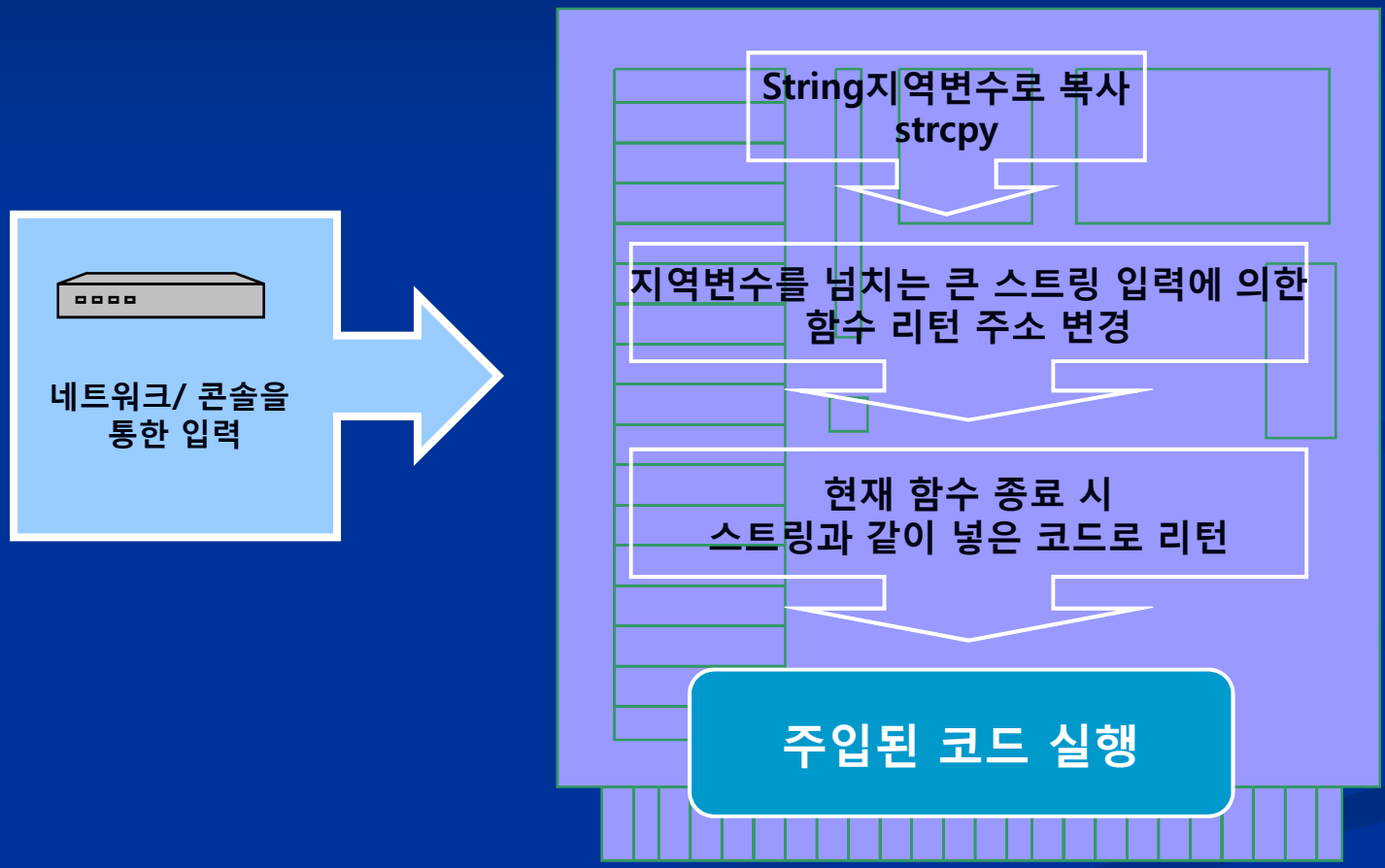

가상주소(Virtual Address)
Page Index
offset
Physical Address 0
Physical Address 1
Physical Address 2
Physical Address 3
Page Table Entry
Page 0
Page 1
Page 2
Page 3
Page 4
Page 5
Page 6
Page 7
실제 메모리

4KByte Page 주소변환 예



Buffer Overflow

개 요


네트워크/ 콘솔을
통한 입력
String지역변수로 복사
strcpy
지역변수를 넘치는 큰 스트링 입력에 의한
함수 리턴 주소 변경
현재 함수 종료 시
스트링과 같이 넣은 코드로 리턴
주입된 코드 실행

원 리

터미널,네트워크를 통한 코드주입

Dummy글자 (40 byte) +
Return주소 (0xD030) +
Dummy글자(4 byte) +
주입할 코드 (? byte)

Tv_Str 변수를 넘쳐
리턴주소등을 덮어씀

STACK

내 용	크 기	주 소
…		
Tv_Str	32	0xD000
지역변수 2	4	0xD020
지역변수 1	4	0xD024
리턴 주소	4	0xD028
함수 파라매터	4	0xD02C
이전함수영역	?	0xD030
…		

실행 조건

- 펌웨어 업그레이드가 자주 없어 야한다.
- Paging에 의한 프로세스가 독립되어야 유리하다.
- 최소의 코드로 목적 하는 바를 이룰 수 있는 환경이 구축되어있어야 한다.
- 리버스 엔지니어링에 쉽게 노출된 제품일수록 약점을 찾기 쉽다.

방지 대책

- 외부로부터 문자열등을 입력 받을 때에는 최대 입력길이 제약을 주어야 한다.
- 외부 네트워크를 통한 터미널등 지원하지 않도록 커널을 설정한다.
- 외부로 부터 들어오는 텍스트정보를 되도록 지역변수에 넣지 않는다.

펌웨어 보호의 다른 위험요소

위험요소와 대책

항 목	내 용	대 책
복 제	타회사에서 막대한 개발비를 들여 개발한 제품을 무단으로 복사. 주로 플래시를 복사하는 유형	하드웨어와 공조하여 GPIO등에 킷값을 숨김 FPGA를 사용
데이터 노출	저장된 자료나 네트워크로 전송중인 자료를 유출 (ex. CCTV ,Network Storage)	외부전송 네트워크 패킷의 암호화 저장장치의 암호화

리버스 엔지니어링과 임베디드

임베디드 개발에서 효과

- 기술 습득 경로의 다양화
- 컴파일러의 버그 검출
- 제품의 고속화
- 제품의 안정화

감사합니다.

4nsys 송 민 호

CodeEngn

www.CodeEngn.com