시스템 해킹 스터디

과제 (05-1)

구본현

목차

1. 아키텍처란

IBM 7030 이 만들어졌을 때 사용자로부터의 요구를 컴퓨터에 도입하기 위한 설계 기술이란 뜻으로 처음 사용되었다.

현재는 컴퓨터 구조, 소프트웨어, 네트워크등 다양한 분야에서 각 부분의 논리 명제를 다루는 것, 구조, 행위를 나타내는 개념적 모형을 뜻한다.

1-1 마이크로 아키텍처(CPU 아키텍처, 컴퓨터 시스템)

컴퓨터 공학에서 CPU 또는 이와 관련되어 하드웨어의 운영에 대해 세세하게 서술되어있는 것을 뜻한다.

Intel 과 AMD 의 아키텍처 명

	INTEL	AMD
32BIT	i386, i486	-
64BIT	x86-64(intel 64)	AMD 64

서브루틴

함수, 메소드, 서브루틴, 루틴, 프로시저는 전부 컴퓨터의 동작을 수행하는 코드를 뜻하지만 포괄적인 의미는 조금씩 다르다.

2. 함수 호출 규약이란

함수 호출 규약은 서브루틴이 호출자로부터 어떻게 처리를 하는지에 대한 규약이다.

CPU 와 같은 하드웨어나 소프트웨어의 환경에 따라 만들어졌다.

2-1 다수의 아키텍처가 사용중인 c 계열 함수 호출 규약

규약 이름	전달인자(파라미터) 해석	스택 정리	기타
Cdecl	역순으로 스택에 들어감 Func1(int a, int b) 일 때 a=1, b=2 이면 스택에 2, 1 PUSH	가변인자 함수를 지원하기 때문에 끝난 후 ESP 의 위치가 같아야 함 Caller 가 Callee 의 스택을 정리함 ex : <add esp, (size)>	Visual Studio 2015 기준 기본형식임
Stdcall		Callee 가 자신의 스택을 직접 정리함 ex : <ret>	Win 32 API 의 기본형식임
Fastcall	처음 두 인자는 순서대로 ECX, EDX 에 들어가며 나머지 인자는 역순으로 스택에 들어감 Func1(int a, int b, int c, int d) 일 때 a=1, b=2, c=3, d=4, 이면 ECX=1 EDX=2 그리고 스택에 4 와 3 PUSH		64bit 에선 Fastcall(의 확장) 만 사용됨 아직 비표준화임, 컴파일러마다 조금씩 다름

RISC(Reduced Instruction Set Computer)
CPU 명령어를 줄여 하드웨어 구조를 좀 더 간단하게 만드는 방식이다.

3. ARM 함수 호출 규약

ARM 은 RISC 계열의 저전력 아키텍처이다. MIPS 와 다르게 조건부 실행, condition-register 를 가지고 있고 복잡한 어드레싱, 프리 시프트 산술연산 및 저장연산에 특화되어있다.

ARM 의 함수 호출 규약에서 함수의 반환 값을 포함한 인자들은 레지스터 r0, r1, r2, r3 를 통해 전달되며 레지스터가 연속된 메모리 공간처럼 사용된다.

32bit 값일 경우 r0 만을 64bit 일 경우 r1 에 상위 32bit 가 저장, 128bit 경우 r0 부터 r3 까지 사용되며 그 이상일 경우 자동으로 정렬된 스택을 이용한다.

r0~r3 를 제외한 나머지 레지스터가 하는 역할

Register	Synonym	Special	Role in the procedure call standard
r15		**PC**	The Program Counter.
r14		**LR**	The Link Register.
r13		**SP**	The Stack Pointer.
r12		IP	The Intra-Procedure-call scratch register.
r11	v8	FP	ARM-state variable-register 8. ARM-state frame pointer.
r10	v7	**SL**	ARM-state variable-register 7. Stack Limit pointer in stack-checked variants.
r9	v6	**SB**	ARM-state v-register 6. Static Base in PID,/re-entrant/shared-library variants
r8	v5		ARM-state variable-register 5.
r7	**v4**	**WR**	Variable register (v-register) 4. Thumb-state Work Register.
r6	**v3**		Variable register (v-register) 3.
r5	**v2**		Variable register (v-register) 2.
r4	**v1**		Variable register (v-register) 1.
r3	**a4**		Argument/result/scratch register 4.
r2	**a3**		Argument/result/ scratch register 3.
r1	**a2**		Argument/result/ scratch register 2.
r0	**a1**		Argument/result/ scratch register 1.

4. MIPS 함수 호출 규약

MIPS 는 RISC 계열의 아키텍처이다. 단순한 명령어 체계를 자랑하며 ARM 과 다르게 세가지 종류의 명령어로 구성되어있다.

MIPS 의 함수 호출 규약에서 함수의 반환 값을 포함한 인자들은 레지스터 r0, r1, r2, r3 를 통해 전달되며 레지스터가 연속된 메모리 공간처럼 사용된다.

타입별 특징

타입	설명
R	rs 와 rt 를 산술연산 하여 rd 에 저장함
I	rs 와 imm(임의의 상수)를 산술연산하여 rt 에 저장함
J	분기문, 이동할 메모리 주소를 연산에 사용함

타입별 구성

Type	-31-			format (bits)		-0-
R	opcode (6)	rs (5)	rt (5)	rd (5)	shamt (5)	funct (6)
I	opcode (6)	rs (5)	rt (5)	immediate (16)		
J	opcode (6)	address (26)				