08장 서브루틴과 제어 추상화

【8.2.3】 레지스터 윈도우

예 8.63
Sparc상의 레지스터 윈도우

서브루틴 호출과 복귀에서 레지스터를 저장하고 복구하는 것에 대한 대안으로 초기 버클리 RISC 기계[PD80, Pat85]는 레지스터 윈도우라 하는 하드웨어적 방법을 구체화시켰다. 기본 생각은 아주 큰 집합의 물리적 레지스터를 제공하고 그 중 대부분을 부분적으로 중복된 윈도우(◎심화학습에 있는 그림 8.10)의 집합으로 구성하는 것이다. 일부 레지스터 이름(그림에서 r0~r7)은 항상 동일한 위치를 참조한다. 하지만 나머지(그림에서 r8~r31)를 현재 활성 중인 윈도우에 관련해 해석한다. 서브루틴 호출에서 하드웨어는 별개의 윈도우를 이동시킨다. 매개변수의 전달을 편리하게 하기 위해 이전과 새 윈도우는 부분적으로 중복된다. 호출 윈도우의 상위 일부 레지스터(그림에서 r24~r31)는 피호출자의 윈도우의 하위 일부 레지스터(그림에서 r8~r15)와 동일하다. 레지스터 윈도우를 사용하는 기계에서 컴파일러는 윈도우의 중간 부분에 현재 서브루틴 내에서만 사용하는 값을 위치시킨다. 호출된 루틴에 값을 전달하기 위해 윈도우의 상위 부분에 그 값을 복사한다. 피호출자 내에서는 자신의 윈도우의 하단 부분에서 그 값을 읽는다.

물리적 윈도우의 수를 고정시키기 때문에 서브루틴 호출의 긴 체인은 하드웨어가 레지스터의 끝까지 다 써버리게 한다. 이는 운영체제에 프로세서를 인도하는 "윈도우 오버플로우" 인터럽트를 유발시킨다. 그 인터럽트 처리기는 이용 가능한 윈도우의 집합을 순환 버퍼 같이 다룬다. 그것은 한 개 이상의 윈도우의 내용을 메모리에 복사하고 나서 실행을 재개한다. 나중에 "윈도우 언더플로우"는 제어가 자신의 콘텐츠를 메모리에 써넣은 윈도우로 복귀하려 할 때 발생할 것이다. 다시 운영체제는 저장된 레지스터를 복구하고 실행을 재개함으로써 회복된다. 실제로 8개의 윈도우는 일반적인 프로그램상에서 오버플로우와 언더플로우를 상대적으로 드물게 만드는 것을 만족시키기 위해 나타난다.

RISC 프로세서에서 레지스터 윈도우를 사용해왔다. 하지만 이 중 단 하나, Sparc은 오늘날 상업적으로 중요하다. 좀 더 최신의 Intel IA-64(Itanium)도 RISC 기계가 아니지만 레지스터 윈도우를 사용한다. 물론 윈도우의 이점은 전형적인 서브루틴 호출에 필요한 불러오기와 저장하기의 수를 경감시키는 것이다. 동시에 레지스터 윈도우는 현재 실행 중인 프로그램과 관련된 상태의 양을 급격히 증가시킨다. 운영체제가 잠시 동안(대부분의 시스템이 1초마다 많은 횟수를 하는 것) 다른 애플리케이션으로 프로세서를 넘기기로 결정하면 그것은 모든 상태를 메모리에 저장하거나 새로운 프로세스가 저장되지 않은 윈도우에 접근하려 시도하면 프로세서를 OS로 트랩하게 한다. 더 나쁘게도 레지스터 윈도우가 제어의 단일 스레드의 참조 환경을 훌륭하게 저장하지만은 2개 이상의 참조 환경(실행 문맥)을 필요로 하는 언어에 대해서는 그렇게 잘 동작하지 않는다. 재개(6.2.2절), 반복자(6.5.3절)과 동시 실행 루틴(8.6절)을 포함한 일부 언어의 특징은 레지스터 윈도우를 사용하는 기계에서 구현하기 어렵다. 왜냐하면 문맥 전환을 할 때 보이는 레지스터뿐만 아니라 다른 윈도우의 그것도 저장하고 복구하게 요구한다. 특히 매개변수에 대해 불러오기와 저장하기가 거의 항상 캐시에 적중한다고 하면 서브루틴 호출 오버헤드에 있어서 경감이 일반적인 애플리케이션의 작업 부하의 추가적인 비용보다 중요한지는 불확실하다.

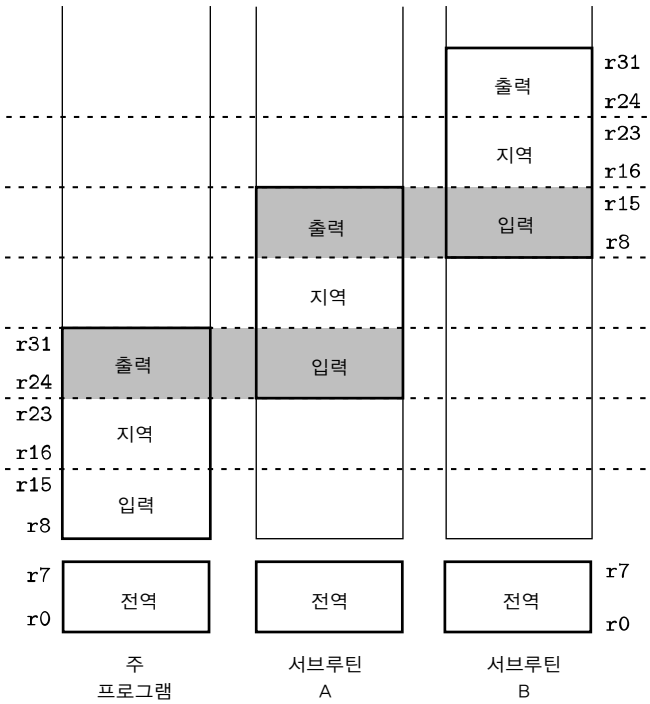

그림 8.10 | 레지스터 윈도우. 주 프로그램이 서브루틴 A를 호출하고 다시 A가 B를 호출하면 레지스터 이름 r0~r7은 동일한 위치를 계속 참조하고 있다. 하지만 새로운, 중복된 윈도우를 참조하게 레지스터 이름 r8~r31을 변경한다. 호출자에서 높은 숫자를 가진 레지스터는 피호출자에서 낮은 숫자를 가진 레지스터와 위치를 같이 쓴다.

✓ 확인문제

56. 레지스터 윈도우는 무엇인가? 어떤 목적을 위해 사용하는가?

57. 어떤 상업적인 명령어 집합은 레지스터 윈도우를 포함하는가?

58. 레지스터 윈도우의 오버플로우와 언더플로우에 대한 개념을 설명하라.

59. 왜 레지스터 윈도우는 다중 스레드 프로그램에 대해 잠재적인 문제인가?