火曜・ピラミッド・ハスの花

ムラの長老は、ピラミッドの石段に腰かけて、夕焼けを見ていた。

「この石は、誰が最初に運んだのだろう。千人か、百人か、それとも一人か。答えは、ナイルの水だけが知っている。」

ナイルの水は、毎年、ムラの畑を越えてやってきた。水が引くと、畑には黒い土が残った。その土に、ハスの種を蒔く。芽が出て、葉が伸び、花が咲く。ハスの花は、朝、開き、夕方、閉じる。

ある日、ピラミッドの石段の一番下に、一輪のハスの花が咲いた。誰が、いつ、どうやって、その花を石段の隙間に挿したのか、誰も知らない。

ムラの子どもたちは、その花を摘んで、頭に挿した。長老は、その花を摘まなかった。

「花は、摘まれると、枯れる。」

長老は、言った。

ピラミッドは、何のためにあるのか。王の墓だ、という。王は、死んだ。死んだ王は、ピラミッドの中に、眠っている。

ムラの人は、ピラミッドを、遠くから、見た。近くには、行かない。

ある夜、長老は、夢を、見た。

夢の中で、長老は、ピラミッドの、頂上に、立っていた。

頂上には、一本の、ハスの、花が、咲いていた。

長老は、その、花を、摘んだ。

長老は、目を、覚ました。

長老は、その、夢を、誰に、話さなかった。

翌朝、長老は、ムラの、畑を、歩いた。

畑の、黒い、土の、上に、ハスの、芽が、出ていた。

長老は、その、芽を、一つ、摘んだ。

長老は、その、芽を、ピラミッドの、石段の、隙間に、挿した。

長老は、ピラミッドを、見た。

ピラミッドは、何も、言わない。

ピラミッドは、何も、知らない。

ピラミッドは、ただ、そこに、ある。

ムラの、人は、ピラミッドを、遠くから、見た。

ムラの、人は、その、芽を、見た。

ムラの、人は、何も、言わない。

ムラの、人は、ただ、畑を、耕した。

ムラの、人は、ただ、種を、蒔いた。

ムラの、人は、ただ、水を、待った。

ナイルの、水は、毎年、ムラの、畑を、越えて、やってきた。

ナイルの、水は、何も、言わない。

ナイルの、水は、ただ、流れた。

ムラの、人は、ただ、生きた。

ムラの、人は、ただ、死んだ。

ムラの、人は、ただ、生まれた。

ムラの、人は、ただ、生まれた。

ムラの、人は、ただ、生まれた

上下文

上下文是人类理解语言的基础。一个词的意义，一个句子的含义，一段话的目的，都需要放在更大的语境中才能被准确理解。这个语境可以是前面说过的话，也可以是说话时的场景、说话人的身份、听话人的预设、共同的背景知识、甚至是整个文化传统。

语境的层次

语境可以分成几个层次。最近的是上文，紧接着说话的那几句话。再远一点是同一篇文章里的其他部分。再远是说话时的物理环境和社会环境。最远是说话人和听话人共同拥有的文化背景、价值观、知识体系。

共同背景

两个人能顺利交流，前提是他们有大量的共同背景。这背景大到连自己都没意识到。比如说中国人说“吃了没”，这不是问吃饭，是问对方现在状况如何。外国人学了这句话，照字面翻译，就闹笑话了。

省略

语境之所以重要，是因为人说话时大量省略。能从上下文推出来的，就不说出来。这样说话才经济。比如说“我昨天去了北京，今天还要去”，第二个“去”后面的“北京”就省略了。听话人能补上，因为上下文提供了信息。

歧义

没有语境，很多话就有歧义。“他来了”，谁来了？“我喜欢这个”，这个是什么？“明天见面”，见面的地点是哪？这些问题都要靠上下文来解决。语境把模糊的东西变清楚。

文化

语境里最深的是文化。同样的话，在不同文化里意思可能完全不同。直接说“不”在某些文化里是礼貌的拒绝，在另一些文化里是粗鲁的冲撞。这些差别不是语言学能解决的，要靠长期的文化浸润。

翻译

翻译最难的就是处理语境。把一句话从一种语言翻成另一种语言，光换词不够，还要换语境。比如说英语的“How are you”翻成中国人说的“吃了没”，这不是字面对应，是两种文化里问候方式的对应。

结语

上下文是人类交流的隐形骨架。看不见摸不着，但没了它，话就说不成。学会理解语境，是成熟的标志。

명령어 체계

1 (구) 명령어 체계 — 단일 문자 1바이트 명령어 ...

단·명령어 체계는 모든 명령어가 한 바이트로 구성된다. 이 방식은 명령어 종류가 256개를 넘지 않는 한 디코딩 속도가 매우 빠르다. 반면 피연산자 주소는 별도 바이트를 요구하므로 메모리 접근 횟수가 늘어난다. 초기 설계에서는 이 트레이드오프를 허용 가능한 수준으로 판단했다.

2 (현) 가변 길이 명령어 — 1~3바이트 ...

명령어 길이는 첫 바이트의 상위 2비트로 결정된다. 00은 1바이트, 01은 2바이트, 10은 3바이트 명령어다. 11은 예약되어 있으며 향후 확장 시 4바이트 명령어에 사용될 수 있다. 디코더는 첫 바이트를 읽은 즉시 나머지 바이트 수를 계산하여 다음 사이클에 페치를 시작한다.

레지스터 파일

레지스터 파일은 16개의 범용 레지스터(R0~R15)와 1개의 플래그 레지스터(FLAG)로 구성된다. R0은 하드웨어 상 특별한 의미가 없으나 관례상 함수 반환 값을 담는다. R14는 스택 포인터(SP), R15는 프레임 포인터(FP)로 사용된다. 플래그 레지스터는 Z(zero), C(carry), N(negative), V(overflow) 4비트만 정의되어 있다.

메모리 맵

주소 범위	용도	비고
0x0000 – 0x0FFF	부트 ROM	읽기 전용
0x1000 – 0x7FFF	사용자 RAM	일반 변수
0x8000 – 0x8FFF	스택	아래 방향 성장
0x9000 – 0xBFFF	힙	동적 할당
0xC000 – 0xFFFF	입출력 포트	매핑 레지스터

인터럽트 벡터

1 (구) 인터럽트 벡터 테이블 — 고정 주소 ...

인터럽트 벡터 테이블은 0x0010부터 시작하며 각 벡터는 2바이트 주소를 저장한다. 타이머 인터럽트는 0x0010, 직렬 포트 수신은 0x0012, 외부 핀은 0x0014에 배치되었다. 인터럽트 발생 시 하드웨어는 현재 PC를 스택에 push하고 해당 벡터 주소로 점프한다. 핸들러 종료 시 RETI 명령어가 PC를 복원하고 인터럽트 플래그를 해제한다.

2 (현) 벡터 테이블 재배치 — 부트 ROM 해제 후 ...

부트 ROM 실행 완료 후 벡터 테이블은 0x1000으로 재배치되어 사용자 RAM 영역에 위치한다. 재배치 주소는 MVTR 명령어로 설정한다. 재배치 후에는 부트 ROM 영역이 읽기 전용 메모리로 전환되어 사용자 코드가 실수로 덮어쓰는 것을 방지한다.

부록 B. 용어 집

용어	설명
니모닉	명령어 코드를 사람이 읽기 쉬운 문자열로 표현한 것
오프셋	기준 주소에 더해지는 상대 주소
플래그	연산 결과 상태를 나타내는 비트
벡터	인터럽트 핸들러 시작 주소를 저장한 메모리 위치

附 录