시스템 소프트웨어

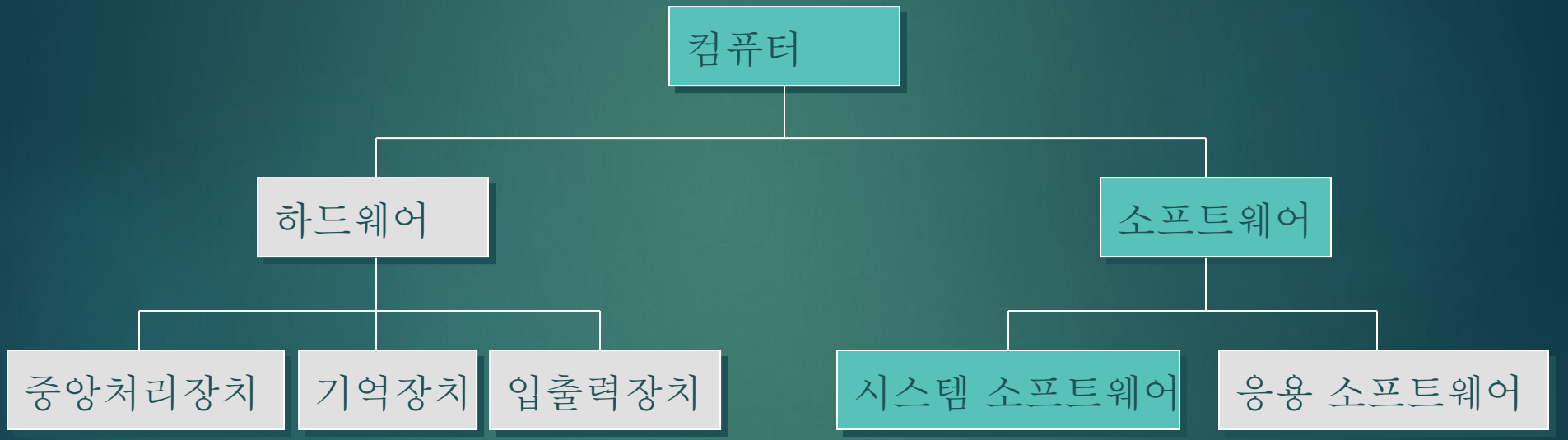
이번 학기 학습 내용

- 시스템 소프트웨어의 개요
- 컴퓨터 동작의 기본 개념
- 인텔 프로세서의 내부 구조
- 8086 어셈블리어(nasm)
- 매크로 프로세서 설계
- 8086 어셈블러 설계
- 링커와 로더 설계

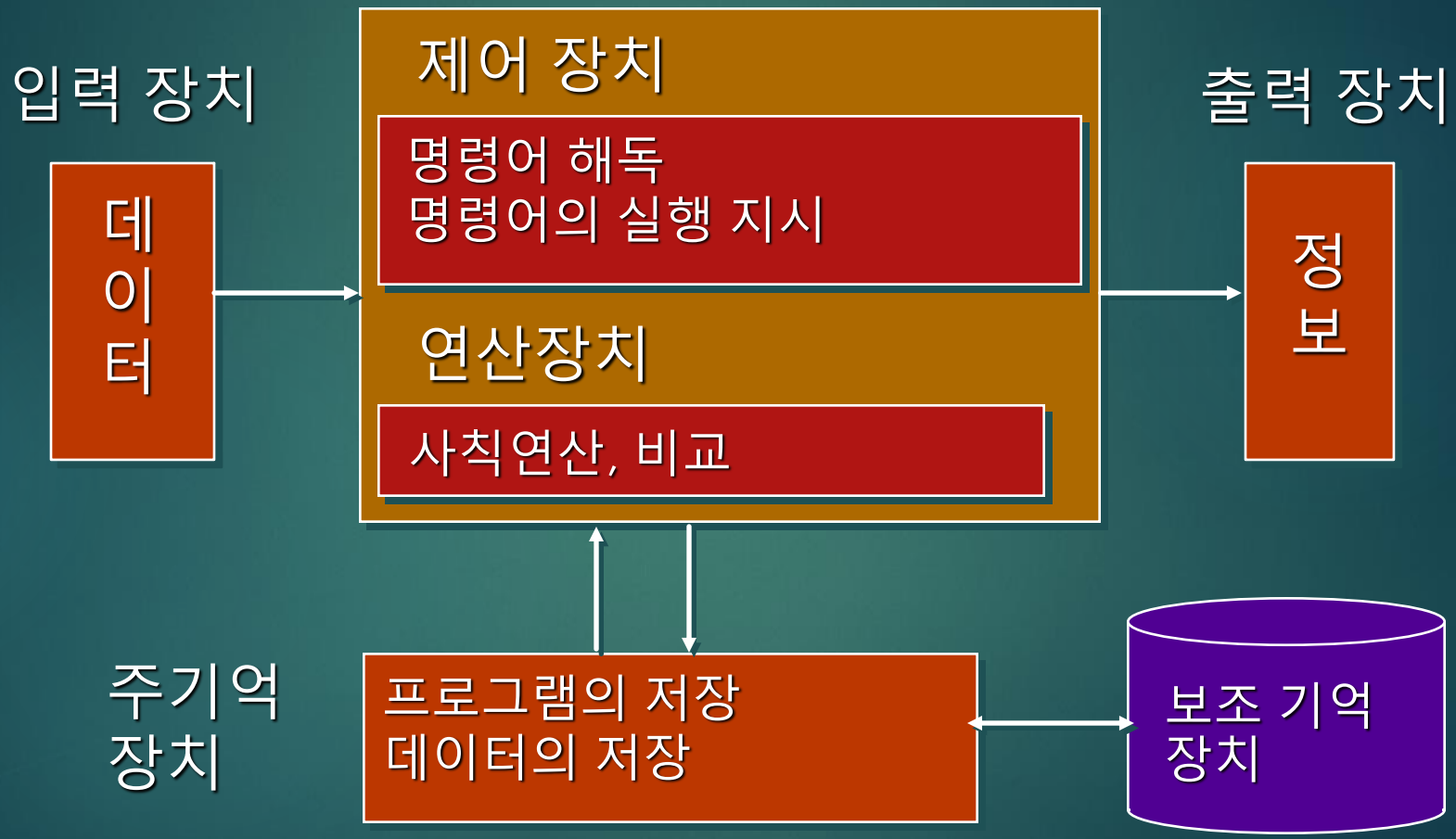
제 1장
시스템 소프트웨어의 개요

- 컴퓨터 시스템 및 하드웨어 구성
- 컴퓨터의 구성과 기능
- 시스템 프로그램의 개요
- 시스템 프로그램의 계층 구조
- 프로그래밍 언어의 계층 구조와 종류
- 어셈블러, 로더, 매크로프로세서
- 컴파일러와 인터프리터, 운영체제

컴퓨터 시스템의 구성



컴퓨터의 구성과 기능


입력 장치
데이터
제어 장치
명령어 해독
명령어의 실행 지시
연산장치
사칙연산, 비교
출력 장치
정보
주기억 장치
프로그램의 저장
데이터의 저장
보조 기억 장치

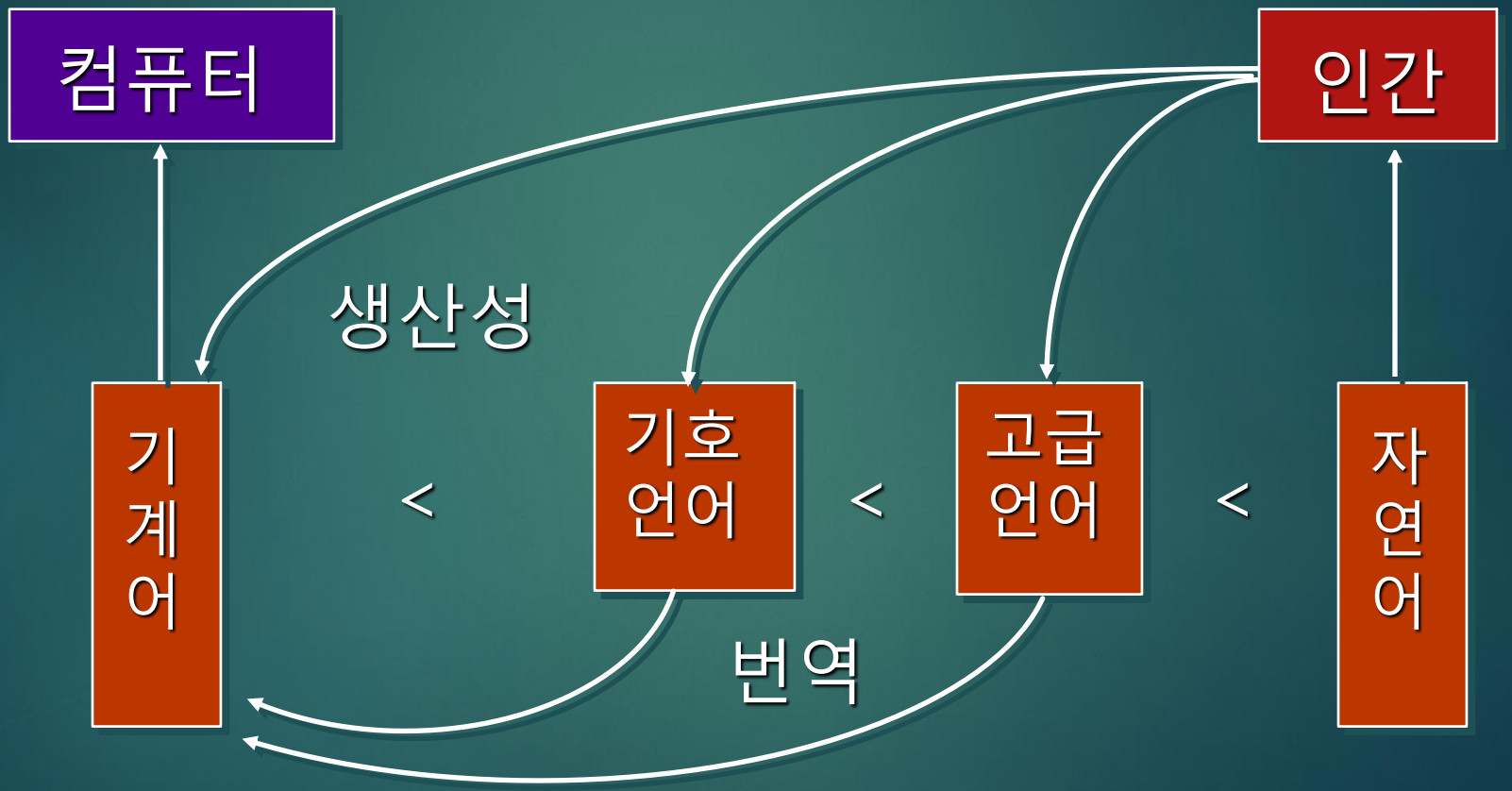
펌웨어

- 읽기 전용 메모리(ROM) 또는 프로그래밍 가능 ROM(PROM)에 저장된 소프트웨어로 하드웨어의 빠른 성능과 견고성 그리고 소프트웨어의 유연성 사이의 균형을 제공. 하드웨어 시스템에 비해 변경이 쉽지만 디스크에 저장된 소프트웨어에 비해 더 많은 노력과 전문 도구가 필요. 펌웨어는 시스템의 초기화, 부트스트래핑, 하드웨어 인터페이스, 시스템 구성 등에서 중요한 역할

프로세서의 개념

- 중앙처리장치의 중요 구성 요소
- 오늘날 마이크로프로세서가 보편적
- 인텔의 80486/586, 펜티엄 프로 등
- 점점 고밀도, 고속, 병렬화
- 전체 시스템을 제어하고 산술, 논리 연산 수행
- 프로세서의 작업 순서를 프로그램

프로그래밍 언어의 계층 구조



프로그램밍 언어의 종류

- 저급언어: 프로세서에 따라 다름

  *기계어, 어셈블리어

- 고급언어: 프로세서에 무관

  * 컴파일러 간에는 호환성이 없다

  *COBOL, FORTRAN, PASCAL, C++

시스템 소프트웨어의 개요

- 하드웨어의 관리
- 마이크로 코드화
- 운영체제, 컴파일러, 유틸리티
- 시스템 소프트웨어를 작성하는 일을 시스템 프로그래밍

시스템 소프트웨어의 계층 구조

시스템 프로그램의 진화 과정

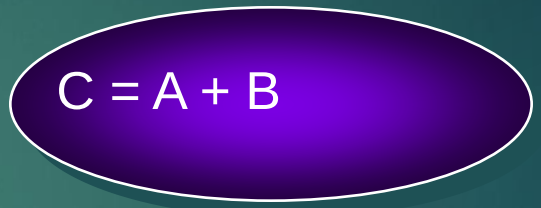
- 어셈블러
- 매크로프로세서
- 링커와 로더
- 트랜스레이터
- 운영체제

어셈블러

- 기계어(machine language)에 대응하는 기호 사용
- 연상(mnemonic or symbol) 기계어라고도 함
- 어셈블리어를 기계어로 번역
- 목적(object) 프로그램의 생성

어셈블리 프로그래밍의 예

고급언어



어셈블리어

```
MOV  AX, A
ADD  AX, B
MOV  C, AX
```

매크로 프로세서

- 프로그램의 일부를 축약화
- 매크로 정의
- 매크로 호출
- 매크로 확장

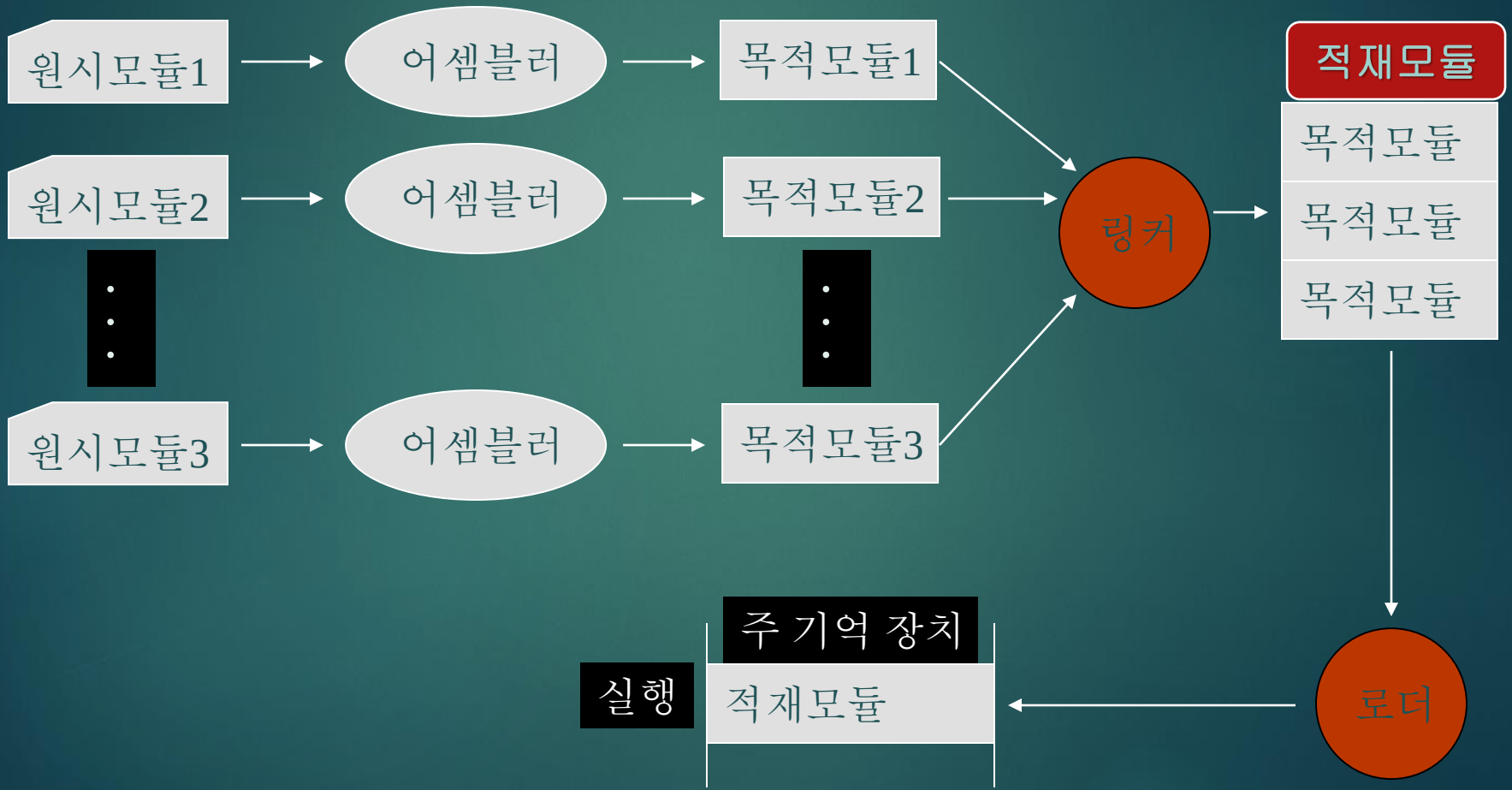
링커와 로더

- 링커
  - 여러 개로 나누어진 원시 모듈들은 어셈블러에 의해 별도로 어셈블 되어 각각 다른 목적 모듈들로 만들어져 보조 기억 장치에 저장
  - 이 모듈들을 실행할 때는 링커로 연결
- 로더
  - 목적 프로그램을 주기억장치에 적재
  - 프로그램의 실행 준비
  - 기억 장소를 적게 점유
  - 절대적 적재와 재배치 적재

링커와 로더

- 어셈블러가 목적 프로그램을 기억장치에 저장하고, 모든 제어 동작하는 기능을 가지면 ?
  - 기억장소 낭비
  - 시간 낭비

링커와 로더


원시모듈1
어셈블러
목적모듈1
원시모듈2
어셈블러
목적모듈2
원시모듈3
어셈블러
목적모듈3
링커
적재모듈
목적모듈
목적모듈
목적모듈
로더
주 기억 장치
실행
적재모듈

절대 적재시 기억장소의 할당

재배치 (relocation)

- 프로그램들이 기억장치 내의 임의의 장소에 적재될 수 있도록 조정
  - 프로그램을 위한 기억 장소 할당(allocation)
  - 목적 프로그램간의 기호적 호출 연결(linking)
  - 주소 상수(address constant)같이 주소에 종속되는 부분을 할당된 기억 장소에 일치하도록 조정(relocation)
  - 실제적으로 기계어 명령들과 자료를 기억 장치에 적재(loading)

컴파일러와 인터프리터

- 컴파일러 : 실행 파일 생성
  - * 일괄 번역형
  - * COBOL, FORTRAN, PASCAL, C++, C
- 인터프리터
  - * 줄당 해석형
  - * BASIC, PROLOG, java, python

운영체제의 개요

- 프로그램의 실행을 제어하는 소프트웨어로 자원의 할당, 스케쥴링, 입출력 제어, 데이터 관리 등의 서비스를 제공하는 것
- 주요 기능

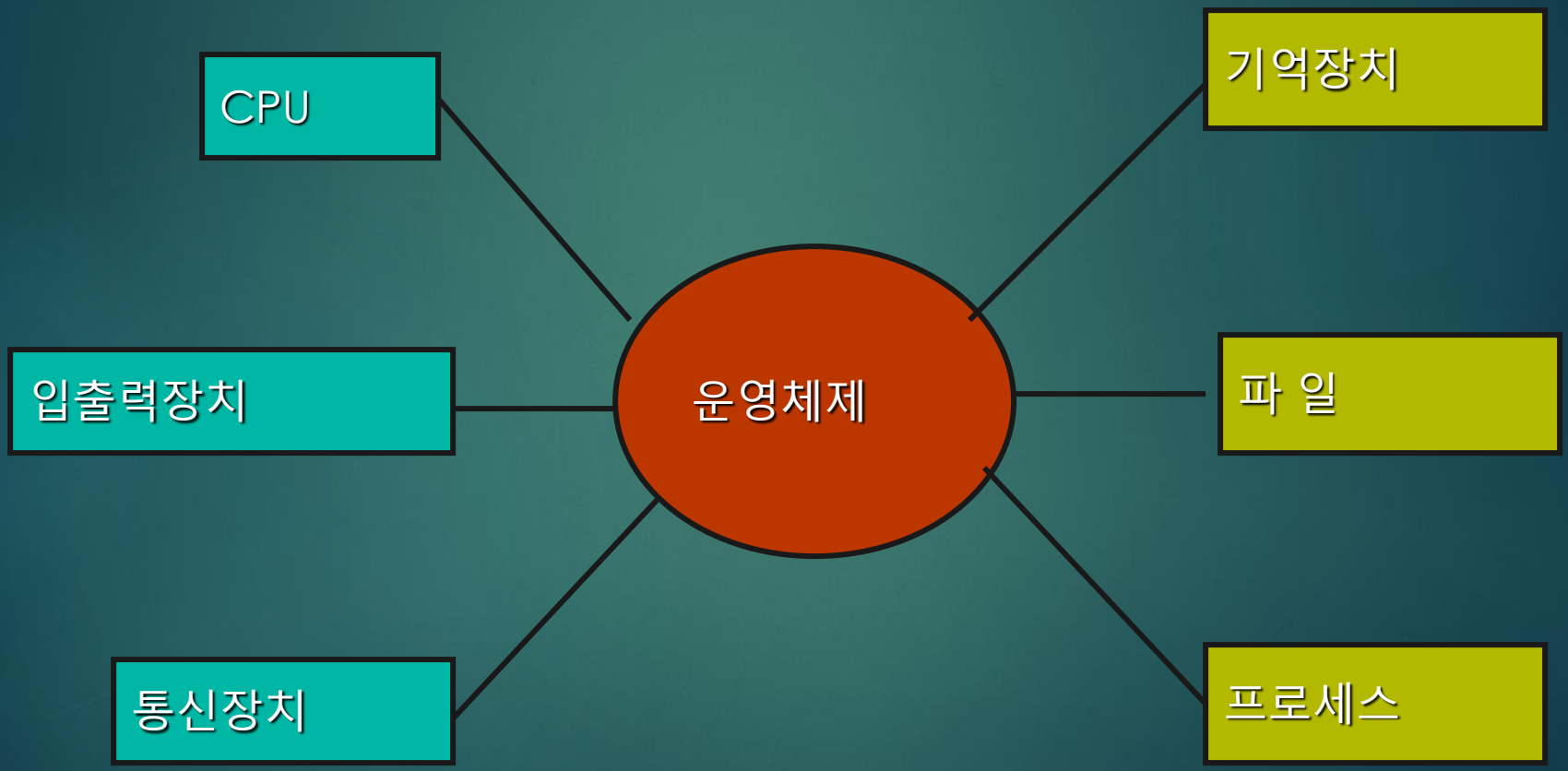
  File Management – 입/출력 및 데이터 관리

  Job Management – 자원 할당(memory, CPU,..), Job 스케줄링

  Task Management - Job을 처리 기본 단위인 Task로 실행
- 분산, 병행, 다중 처리로 발전

  초기에는 일괄, 다중, 시분할 처리

운영체제의 자원 관리



임베디드 시스템의 개요

- 마이크로프로세서 혹은 마이크로컨트롤러를 내장하여 (embedded) 원래 제작자가 지정한 기능만을 수행하는 시스템
- PC는 임베디드 시스템이라 하지 않음
- 설계, 개발 시 고려사항

  Real-Time, Reactive : 실시간

  Small Size, Low Weight : 작은 크기와 무게(휴대용)

  Safe, Reliable : 안정성과 신뢰성

  Harsh Environment : 제한 적인 상황에서도 동작

  Cost Sensitivity : 최소한의 비용으로 높은 효율이 필요

임베디드 시스템의 활용

- 정보 가전 제품
- 핸드폰 및 PDA 단말기
- 공장 자동화 및 자동제어
- 첨단 특수 분야

요약

- 하드웨어와 소프트웨어의 개념
- 프로그램밍 언어
- 시스템 프로그래밍에 대한 이해
- 번역 관련 소프트웨어의 개념

  *어셈블러, 컴파일러, 매크로 프로세서
- 링커와 로더
- 운영체제의 발전과 기능 이해
- 임베디드 시스템의 개요 및 활용