

## 9주차 3차시 패키징

### 【학습목표】

1. 패키징에 대한 개념과 패키징을 하는 이유를 나열할 수 있다.
2. 분석 이후의 패키징을 두 가지로 나누어 구분하여 설명할 수 있다.

### 학습내용1 : 패키징

#### 1. 패키징(packaging)

1) 설계단계에서 작성된 구조도(structured chart)를 구현 가능한 작업(job) 단위로 세분화거나, 컴퓨터에 적재(load) 가능한 단위로 분할하여, 실제로 구현하기 적합한 형태로 구분하여 정리하는 행위임

#### 2. 패키징을 하는 이유는 다음과 같음

- 1) 기억장치의 용량제한으로 모든 모듈을 일시에 적재하기 불가능한 경우
- 2) 적재 가능해도 기억장치의 비효율적 사용, 처리 속도에 문제가 있을 경우
- 3) 서브시스템, 모듈의 경우 시큐리티·복구·감사에 별도 조치가 요구될 경우
- 4) 입출력 데이터에 비동기적 관계가 있을 경우
- 5) 서브시스템 사이에 처리 주기가 다를 경우

#### 3. 패키징에서 사용되는 주요 개념

- 1) 시스템(system)
  - 하나 혹은 여러 개의 응용에 연관되는 「작업(job)」으로 구성된 집합체임
- 2) 작업(job)
  - 한 개의 「작업단계(job phase)」 혹은 「1~n개」의 「작업단계」로 구성된 집합체임
- 3) 작업단계(job phase)
  - 하나의 메인 프로그램을 의미함
  - 하나의 메인 프로그램은 「1~n개」의 서브프로그램으로 구성됨
- 4) 프로그램(program)
  - 서로 기능이 관련된 「1~n개」의 모듈로 구성됨

5) 적재단위(load unit)

- 시스템이 수행되기 전에 호출(call)로 관계 지워진 「1~n개」의 프로그램으로 구성됨

**4. 패키징 시에 준수해야 할 원칙**

- 1) 다수의 모듈에 공유되는 모듈은 분리시켜 별도로 패키징 함
- 2) 논리적으로 순차적으로 처리되는 모듈은 하나로 통합하여 패키징 함
- 3) 예외처리에러처리 기능은 별도로 분리하여 전용모듈로 패키징 함
- 4) 호출빈도가 높은 모듈은 하나로 통합해서 별도의 모듈로 패키징 함

**5. 패키징 시기**

- 1) 분석 이후의 패키징
- 2) 설계 이후의 패키징

**학습내용2 : 분석 이후의 패키징**

**1. 분석 이후의 패키징은 다음 두 가지가 있음**

- 1) 시스템에서 작업 단위로 패키징
- 2) 작업에서 작업단계로 패키징

**2. 시스템(system)에서 작업(job) 단위로 패키징**

- 1) 일반적으로 「시스템을 작업 단위로 분리시키는 패키징」 - 자료흐름도(DFD)의 실제적인 경계를 고려하여 수행해야 함
  - 이때 고려사항은 다음과 같음
    - 하드웨어
    - 처리 환경
    - 처리주기

※ 고려사항

- 하드웨어 : 초기설비비, 운영관리비, 기타비용 등의 측면을 고려하여 최상의 방안을 결정해야 함 (시스템 구성이 다양하기 때문임)
- 처리 환경 : 시스템의 내용·성격을 파악하여 온라인화 여부, 관련업무 파악, 즉시·일괄 처리 등을 명확하게 결정해야 함
- 처리주기
  - (a) 처리주기가 다르면 별개로 패키징함
  - (b) 처리주기가 동일해도 내용·성격이 다르거나 두 가지 유형의 작업이 동시에 발생하면 별개로 패키징 함
- 2) 분석 이후 시스템을 작업 단위로 패키징 하는 장점
  - 자료흐름도(DFD)는 시스템을 작업(job) 단위로 분리하는데 유용한 도구임
  - 자료흐름도는 각 작업 사이에 「자료흐름」을 명확히 파악 가능함
  - 분석을 위해서 사용자(user)와 자주 접촉한 관계로 분석 직후에는 사용자로부터 정보획득이 용이한 시점임

### 3. 작업(job)에서 작업단계(job phase)로 패키징

- 1) 「작업(job)」 단위로 분할한 내용을 「작업단계(job phase)」로 분할하는 경우
- 2) 작업에서 작업단계로 분할하는 과정에서
  - 시스템 관리에 필요한 사항
    - 시큐리티·복구·감사·예비 등의 내용을 별도의 작업단위(job phase) 혹은 중간파일로 만드는게 효과적 일 경우가 있음. 이들을 별도의 작업단위 혹은 중간파일로 작성 여부 판정이 요청됨.
  - 요구되는 컴퓨터 자원
    - 하나의 작업단위(job phase)로 처리 시에 컴퓨터 자원적 제한으로 작업단위로 만드는 것이 효과적일 경우가 있음. 이럴 경우에 추가적인 컴퓨터 자원 요구 여부 판정이 필요함.
  - 기존 패키지 이용문제 등을 고려하여 패키징 함.
    - 기존 패키지는 속성상 별도의 작업단계(job phase)로 구분하여 사용함이 바람직함.

### 학습내용3 : 설계 이후의 패키징

1. 구조도 작성 완료 후에 패키징은 구조도 내용을 「프로그램」 혹은 「적재단위(load unit)」로 분할하는 일임
2. 대체적인 관점에서 구조도의 모든 모듈은 별개의 프로그램으로 작성함이 이상적임
3. 그러나 언어의 종류에 따라서는(COBOL)
  - 1) 하나의 모듈이 하나의 단위 프로그램 의 일부분으로 작성되어 호출하는데 문제가 발생하는 경우가 있음
  - 2) 이러한 문제 해결방안으로서 구조도 에서 빈번하게 호출되는 모듈들을 하나의 「적재단위」로 만듦

4. 구조도의 모듈 중에서 하나의 적재단위로 만들 대상 모듈의 선정기준은 두 모듈 사 이의 「호출빈도」 임 그 「선정기준」은

- 반복호출 모듈
- 상호호출이 빈번한 모듈
- 호출간격이 짧은 모듈

### 【학습정리】

1. 패키징에 대하여 이해한다.
2. 분석 이후의 패키징을 알아본다.
3. 설계 이후의 패키징을 파악한다.