

운영체제_[2020-2학기] Project-2

[Deadlock Detection]

2020.11.02

엄영익

목차

- 프로젝트 소개
 - 목표
 - 세부 내용
 - 구현
 - 제출물 및 제출기한

Project-2

- 목표
 - Deadlock detection 기법 구현
(개인별 프로젝트)
- 세부 목표
 - N개의 process와 M개의 resource type을 갖는 시스템에서 주어진 request matrix와 allocation matrix를 기준으로 deadlock detection을 수행하는 기법 구현
 - N과 M 값은 입력으로 주어지며 request matrix와 allocation matrix도 입력으로 주어짐

Project-2

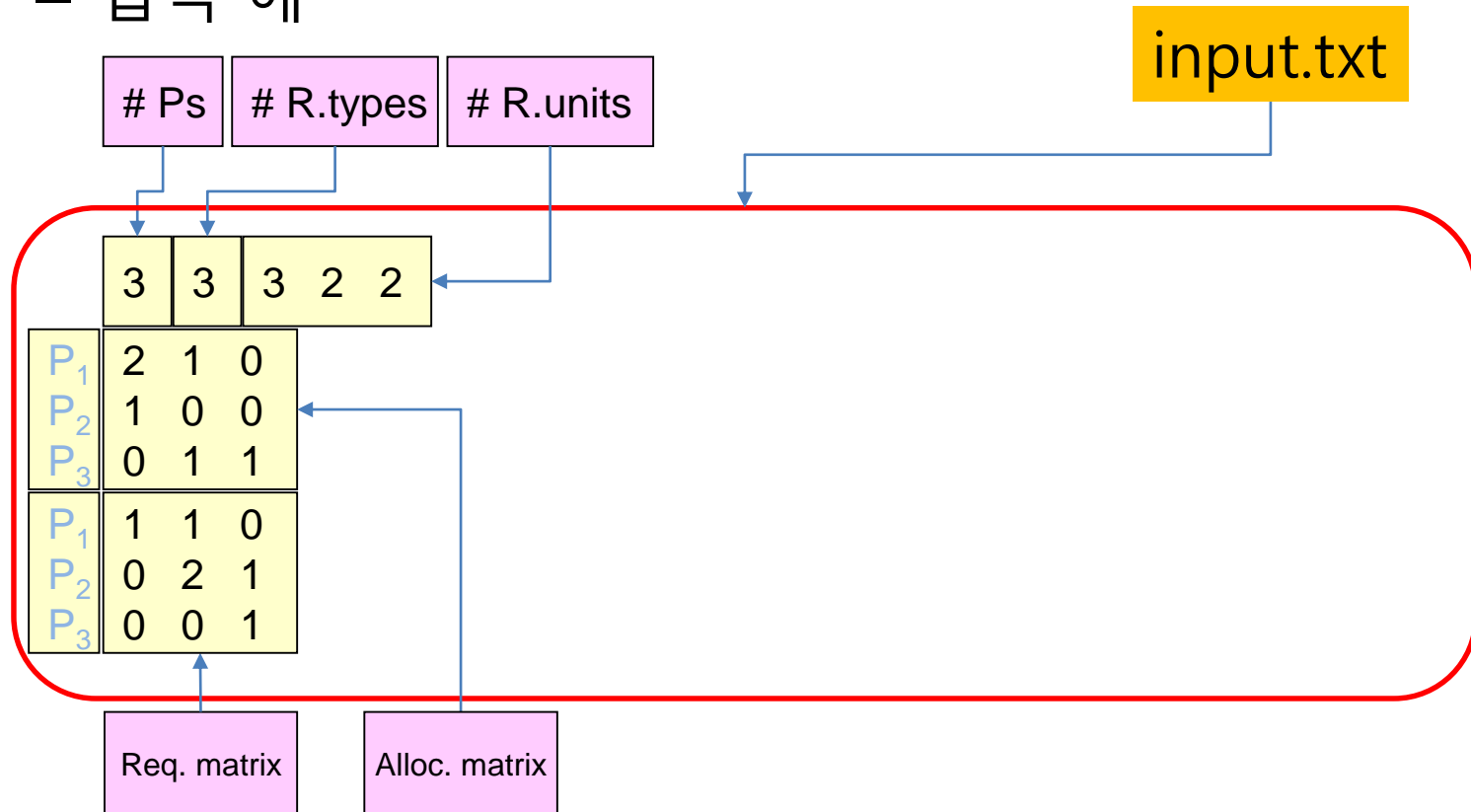
- 세부 내용
 - Graph reduction 기법으로 deadlock detection
 - Multiple resource types & multiple resource units 가정
 - Deadlock detection 결과로
현재 시스템에 deadlock이 존재하는지 여부 및
deadlocked process list를 출력

Project-2

- 구현
 - 입력 (입력 파일명 input.txt로 작성)
 - 프로세스 개수 (N) 및 resource type 개수 (M),
각 resource type 별 resource unit 개수 (t_1, t_2, \dots, t_M)
 - Allocation matrix ($N \times M$ matrix)
 - A_{ij} = # of resource units of type R_j allocated to P_i
 - Request matrix ($N \times M$ matrix)
 - R_{ij} = # of resource units of type R_j requested by P_i
 - 출력
 - Deadlock 상태 여부
 - Deadlock 상태인 경우 deadlocked process list

Project-2

- 구현
 - 입력 예



Project-2

- 구현

- 고려 사항

- 기타 고려사항이나 가정이 필요할 경우, 스스로 판단하여 rule을 정하고, 이를 결과 보고서에 명시해야 함
 - 출력 형태는 각자 자유롭게 결정;
단 출력 결과를 누구나 쉽게 파악할 수 있도록 해야 하며,
출력 형태에 대해 보고서에 설명해야 함
 - 다양한 입력을 사용하여 testing 하고, 그 결과를 보여야 함

- 개발 플랫폼

- Linux 시스템에서 C-언어 사용
 - 불가피한 경우 보고서에 사유를 기재하면서 C++/Python 사용 가능

Project-2

- 제출물 및 제출기한
 - 제출물
 - 결과 보고서 파일
 - 설계/구현 내용 설명
 - 다양한 입력에 대한 실행 결과
 - 제출물 내 각 파일에 대한 설명
 - 최종 소스 코드 파일
 - 제출기한
 - 2020.11.08(일) 오후 8:00 까지 (Hard Deadline)
 - 제출물을 'OS2020-2_학번_이름_P2.zip' 으로 압축하여 iCampus 과제 영역에 온라인 제출

Q&A