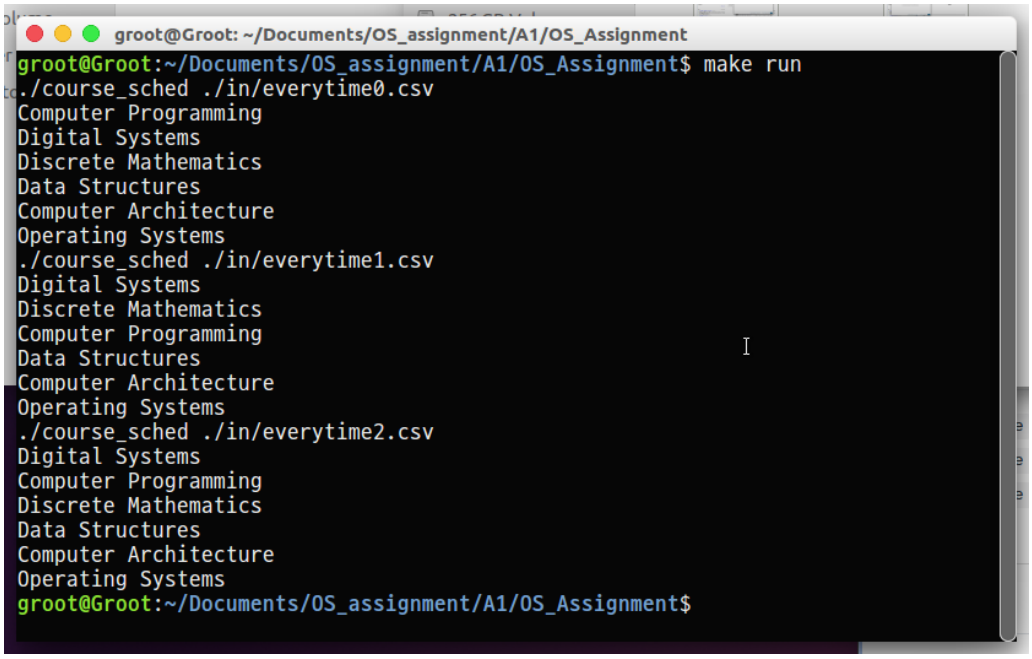
소프트웨어학과 201623405 금도윤

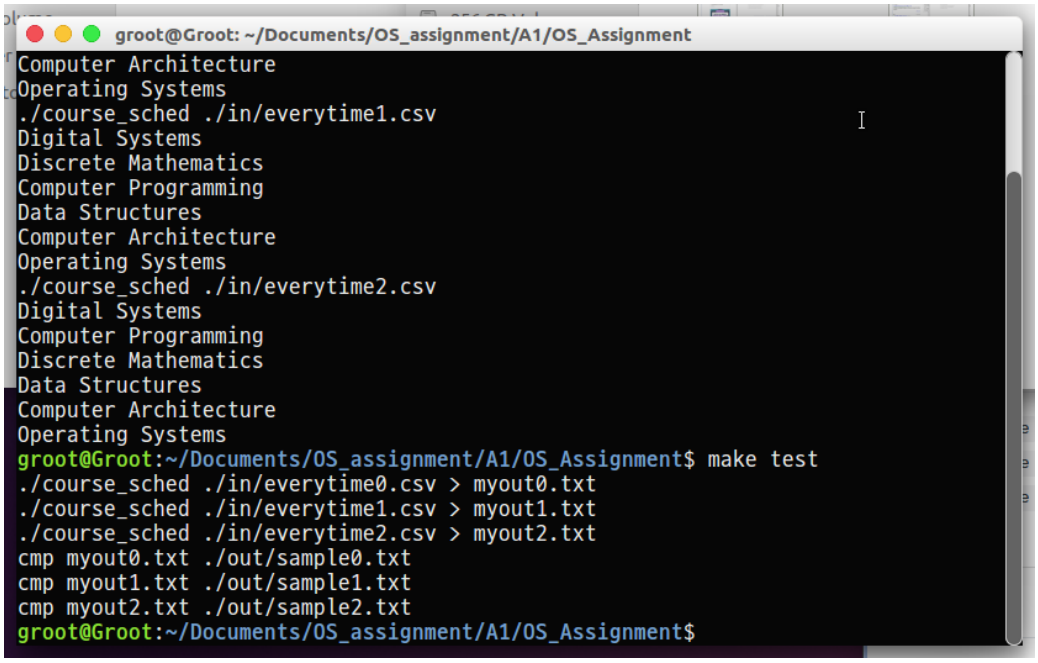
과제의 수행 정도

입력 파일에 있는 database.csv파일을 읽어 필요한 정보를 저장한 후, 강의 평가의 정보가 저장되어 있는 everytime0, everytime1, everytime2의 입력 파일을 토대로 각각 정렬 기준에 맞게 과목들을 출력하였습니다. 또한 출력 결과를 자가 채점하여 주어진 출력 파일과 비교 했을 때 문제가 없이 동일함을 확인하였습니다.

각 입력 파일에 따른 프로그램 실행 결과



자가 채점



cmp 명령어 결과, 출력이 없음을 확인하여 두 파일이 동일함을 확인하였습니다.

과제를 통해 배운 것

- 파일을 읽어 파일에 있는 내용을 의미 있는 data로 추출하는 방법을 익혔습니다.
- 위상 정렬을 위해 그래프를 구상해야 했는데 이때, 간선을 어떻게 만들어야 하나 고민했습니다. 연결 리스트처럼 노드 구조체 멤버에 노드 포인터를 정의하고 이를 이용해 각 노드들을 연결하는 것을 시도했었습니다. 하지만 사이클이 없는 방향 그래프인 것을 깨닫고 이럴 필요 없이 CourseEntry 구조체 포인터를 이용하여 간선을 만들어주면 된다는 것을 깨달았습니다.
- 명령 인자를 이용해서 파일을 읽는 방법을 익혔습니다.
- 정렬을 할 때 어떤 기준으로 먼저 정렬해야 할지 생각하게 되는 시간이었습니다. 선수과목이 적은 순서 > 난이도 높은 순서 > 알파벳 순서로 우선순위가 정해지기에 처음에 난이도와 알파벳 순으로 정렬하고 마지막에 진입차수(indegree)를 이용하여 위상 정렬을 하면 결과가 나온다는 것을 알았습니다. 위상 정렬은 진입차수가 0이 될 때를 이용하기 때문에 이전에 난이도와 알파벳 순으로 정렬한 상태여야 합니다.

과제에 대한 피드백

- 입력 data의 크기가 크지 않아 동적 할당을 하지 않았는데, 과목이 많아 data가 커진다면 동적 할당으로 메모리를 관리하는 것이 좋습니다.
- 난이도와 알파벳 순서로 정렬을 할 때 정렬해야 할 data가 많지 않아, 굳이 더 빠른 시간복잡도를 지닌 병합 정렬이나 퀵 정렬을 사용하지 않았습니다. 더 빠른 정렬 방법을 사용하는 것이 좋습니다.
- database.csv 입력 파일을 읽어 data를 의미 있는 문자열로 자를 때, 기준 문자가 하나로 통일되지 않고 쉼표와 공백 및 개행 문자이기에 어려움을 겪었습니다. 그래서 파싱을 깔끔하게 하지 못 했습니다. 의미 있는 문자열만을 정확히 추출하는 방안을 생각해보는 것이 좋습니다.