

---

Section 5

“**Assignment**”

## 5. Assignment

---

### 공통

#### 3가지 버전의 LRU 알고리즘에 대해 구현 및 시뮬레이션

- [github.com/ku-cloud/22spring-os-lab03](https://github.com/ku-cloud/22spring-os-lab03) 의 assignment/skeleton.c 기반으로 구현
- INPUT: Reference String 길이(50~100), 최대 페이지 번호(5~10), 프레임 사이즈(1~6)
- OUTPUT: Page Fault 발생 횟수

실행 방법 : `$ ./program {ref_arr_sz} {page_max} {frame_size}`

example) `./program 50 10 5`

Reference String은 주어진 ref\_arr\_sz와 page\_max를 기반으로 랜덤 생성

## 5. Assignment

---

### 공통

#### 3가지 버전의 LRU 알고리즘에 대해 구현 및 시뮬레이션

- 과제 결과물 (학번-이름-lab03.zip 으로 압축)
  - 각 버전의 구현 소스 파일 3개 (학번-이름-x.c)
  - 리눅스 환경에서 빌드한 각 버전 별 실행 파일 3개 (학번-이름-x.out)
  - 구현에 대해 간단히 설명한 총 2-3장 내외의 보고서 (학번-이름.pdf)
  - "블랙보드 > 과제 및 시험 > 실습3 과제" 에 제출 바랍니다.

#### - 기간

~ 06.08 10:30am (오프라인 수업에서 답안 예시 공유 예정이므로, 기간 종료 후에는 받지 않습니다)

---

## 5. Assignment

### 공통

#### 3가지 버전의 LRU 알고리즘에 대해 구현 및 시뮬레이션

##### - 보고서에 대한 설명

- (학번-이름.pdf)

- 2-3장 내외

- 각 버전 별 구현에 대해 간단히 설명

- Reference String  $S = \{7, 0, 1, 2, 0, 3, 0, 4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1\}$

에 대한 결과를 오른쪽 그림처럼 출력하여 스크린샷 첨부

(각 버전 별로) (프레임 개수 4로 통일)

```
7 | 7 . . . (fault)
0 | 7 0 . . (fault)
1 | 7 0 1 . (fault)
2 | 7 0 1 2 (fault)
0 | 7 0 1 2
3 | 3 0 1 2 (fault)
0 | 3 0 1 2
4 | 3 4 1 2 (fault)
2 | 3 4 1 2
3 | 3 4 1 2
0 | 3 4 0 2 (fault)
3 | 3 4 0 2
2 | 3 4 0 2
1 | 3 4 0 1 (fault)
2 | 2 4 0 1 (fault)
0 | 2 4 0 1
1 | 2 4 0 1
7 | 2 7 0 1 (fault)
0 | 2 7 0 1
1 | 2 7 0 1
```

## 5. Assignment

---

### Assignment 3-1 (6점)

#### Stack을 이용한 LRU Replacement 시뮬레이션

- 파일명: 학번-이름-1.c

1. **generate\_ref\_arr()** 함수 구현 (랜덤 reference string 생성하여 리턴)

2. **lru()** 함수 구현 (page fault 발생 횟수 리턴)

- 보고서에 다음 내용 추가

- 주어진 Reference String  $S$ 에 대해, 페이지 참조마다 Stack 내용이 어떻게 변하는지 표로 작성

## 5. Assignment

---

### Assignment 3-2 (7점)

#### Clock Algorithm 시뮬레이션

- 파일명: 학번-이름-2.c

1. **generate\_ref\_arr()** 함수 구현 (랜덤 reference string 생성하여 리턴)
2. **lru()** 함수 구현 (page fault 발생 횟수 리턴)

## 5. Assignment

---

### Assignment 3-3 (7점)

#### Additional Reference Bits Algorithm 시뮬레이션

- 파일명: 학번-이름-3.c

1. **generate\_ref\_arr()** 함수 구현 (랜덤 reference string 생성하여 리턴)

2. **lru()** 함수 구현 (page fault 발생 횟수 리턴)

- 보고서에 다음 내용 추가

- 주어진 Reference String  $S$ 에 대해, 페이지 참조마다 0번 페이지의 Reference Bits이 어떻게 변하는지 표로 작성

## 5. Assignment

---

### 추가적인 정보

- skeleton.c 파일은 Git clone해서 받는 것을 권장하지만, 다음 링크에서 내용 복사해가도 무방합니다.
  - <https://github.com/KU-Cloud/22Spring-OS-Lab03/blob/master/assignment/skeleton.c>
- 빌드는 gcc로 진행하며, Linux, macOS 관계 없습니다.
  - 단, Windows 실행 파일(.exe) 제출은 금지합니다.
    - 빌드를 위해 Linux 환경이 필요하다면 클라우드 가상머신을 제공할 수 있으니 조교 메일로 연락바랍니다.
- **lru()** 함수는 input, output만 정확하면 되므로 함수 이름이나 파라미터 이름은 유연하게 수정해도 됩니다.
- 추가 문의 : [freckie@korea.ac.kr](mailto:freckie@korea.ac.kr)