



2016년 2학기 운영체제

# Assignment 3

Dept. of Computer Engineering,  
Kwangwoon Univ.

# Requirements

---

## ▶ Report

- ▶ Introduction : 5줄 이하
- ▶ Reference : 각 과제 별
  - ▶ e.g. 친구 도움, 책, 인터넷 사이트 주소 등
  - ▶ 강의자료만 이용한 경우 생략 가능
- ▶ Conclusion

# Requirements

---

- ▶ **Source**

- ▶ 3-1

- ▶ flvma.c
    - ▶ Makefile

- ▶ 3-2

- ▶ bbp.c
    - ▶ Makefile

- ▶ **Copy 발견 시 0점 처리**

# Requirements

---

## ▶ Softcopy

- ▶ 보고서 및 소스파일은 하나의 압축파일로 압축하여 제출 (tar.gz)
- ▶ 보고서는 PDF로 변환할 것
- ▶ **2016. 11. 25(금) 23:59:59** 까지, [U-캠퍼스] → [온라인참여학습] → [과제제출]
  - ▶ Due Delay 없음

## ▶ Hardcopy

- ▶ 제출하지 않음

# Assignment 3-1:

## Files located in Virtual Memory Area

---

### ▶ Problems

- ▶ 프로세스의 가상 **메모리에는** 파일에서 불러온 여러 정보가 적재되어 있음
  - ▶ e.g. shared library, binary file...
- ▶ 하나의 프로세스에 대하여, 아래의 내용을 출력하는 시스템 콜을 작성
  - ▶ 정보가 위치하는 가상 메모리 주소
  - ▶ 정보의 원본 파일의 전체 경로

# Assignment 3-1:

## Files located in Virtual Memory Area

---

### ▶ Conditions

- ▶ 시스템 콜을 하나 생성하고, 해당 시스템 콜에 wrapping할 것
  - ▶ 시스템 콜 이름: ftrace (\_\_NR\_ftrace, sys\_ftrace)
  - ▶ 시스템 콜 번호: 351번
    - **실습 5주차 강의자료의 시스템 콜 생성 실습 두 개를 모두 진행한 뒤에 본 과제를 진행할 것**
    - 즉, 349, 350번 시스템 콜을 먼저 생성해야 함
  - ▶ 시스템 콜에서 수행할 작업: {return 0;}
  - ▶ **단, 실습수업을 수강하는 학생은 실습의 요구 조건에 따라 구현할 것**
- ▶ 소스 파일 내에는 wrapping을 위한 코드도 포함되어야 함
  - ▶ 실습 6주차 강의자료 참고

### ▶ Hints

- ▶ 관련 강의자료를 참고하여 struct task\_struct 부터 분석을 시작할 것
- ▶ 관련 강의자료의 예시 코드를 참고
- ▶ 코드 분석 툴(ctags, cscope, ...)로 커널 코드와 그에 대한 주석을 살펴볼 것
- ▶ pid\_task()

# Assignment 3-1:

## Files located in Virtual Memory Area

### ► Examples

```
sslslab@oslab:~$ sudo insmod flvma.ko
sslslab@oslab:~$ gcc test.c
sslslab@oslab:~$ ./a.out
sslslab@oslab:~$ dmesg
...
[ 3801.178301] ##### Loaded files of a process 'a.out(9050)' in VM #####
[ 3801.178304] (0x8048000~0x8049000) /home/sslslab/a.out
[ 3801.178305] (0x8049000~0x804a000) /home/sslslab/a.out
[ 3801.178306] (0x804a000~0x804b000) /home/sslslab/a.out
[ 3801.178307] (0x40000000~0x40020000) /lib/i386-linux-gnu/ld-2.15.so
[ 3801.178309] (0x40020000~0x40021000) /lib/i386-linux-gnu/ld-2.15.so
[ 3801.178310] (0x40021000~0x40022000) /lib/i386-linux-gnu/ld-2.15.so
[ 3801.178311] (0x40035000~0x401d8000) /lib/i386-linux-gnu/libc-2.15.so
[ 3801.178312] (0x401d8000~0x401da000) /lib/i386-linux-gnu/libc-2.15.so
[ 3801.178313] (0x401da000~0x401db000) /lib/i386-linux-gnu/libc-2.15.so
[ 3801.178314] #####
sslslab@oslab:~$
```

```
#include <unistd.h>
#include <sys/types.h>
#include <linux/unistd.h>

int main(void)
{
    syscall(__NR_ftrace, getpid());
    return 0;
}
```

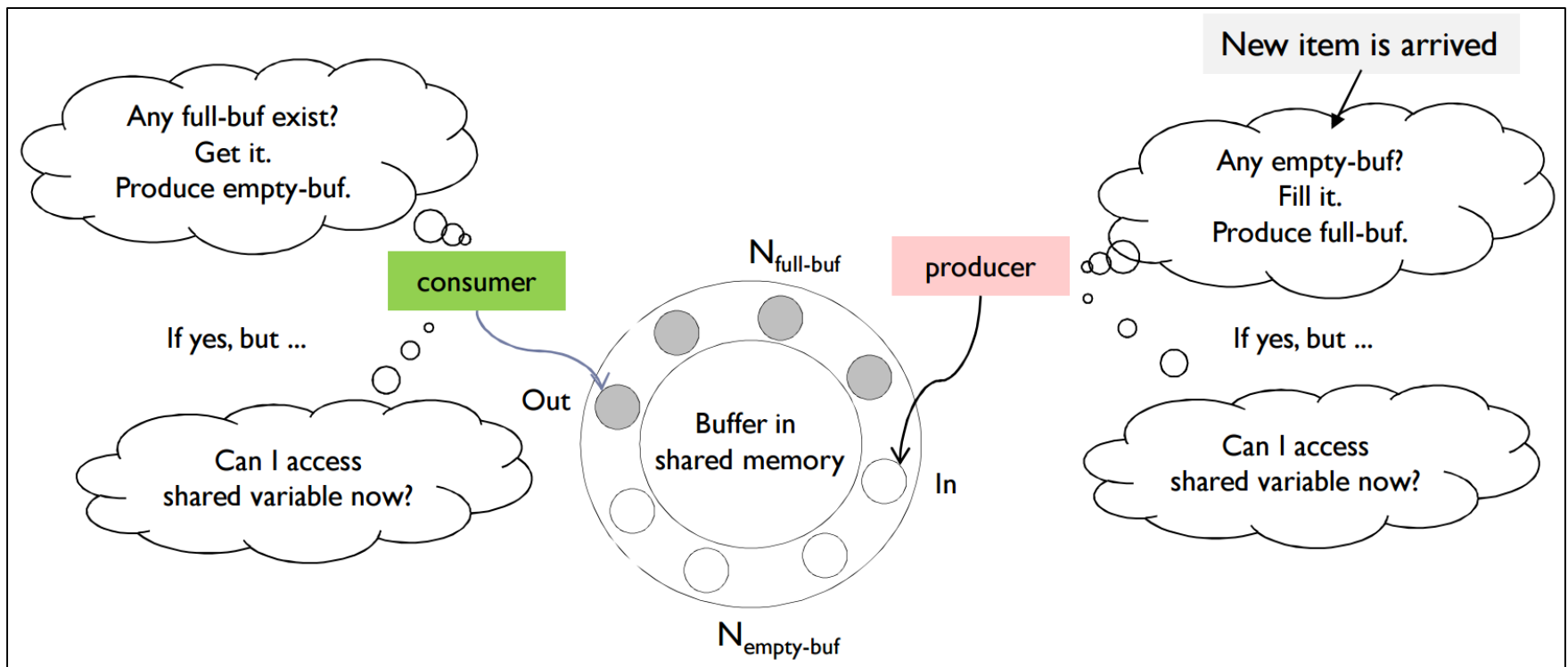
**test.c**

# Assignment 3-2:

## Solution Simulation for Bounded-buffer Problem

### ► Problems

- ▶ 유한한 개수의 데이터를 보관하는 버퍼
- ▶ 생산자와 소비자는 무한히 데이터를 생산 및 소비
  - ▶ 이 때, 다수의 생산자와 다수의 소비자가 접근 가능하도록 동기화 문제를 해결





# Assignment 3-2:

## Solution Simulation for Bounded-buffer Problem

---

### ▶ Conditions

- ▶ Multi-thread 방식, POSIX thread 이용
- ▶ Doubly linked list 형태의 버퍼를 사용하여 구현
  - ▶ 데이터는 다음과 같은 구조체의 형태로 버퍼에 저장

```
struct
{
    int producer_tid;    /* producer's thread ID */
    int data_id;         /* An integer that is incremented by 1 when produced. */
    BUFDATA *prev;       /* previous element in buffer */
    BUFDATA *next;       /* next element in buffer */
} BUFDATA;
```

- ▶ 다음 함수를 제외한 나머지 부분은 자유롭게 구현
  - ▶ producer(), insert() : 데이터 생성, 데이터 삽입
  - ▶ consumer(), delete() : 데이터 추출, 데이터 소비
- ▶ 생산자 및 소비자 thread는 종료 되지 않음
  - ▶ 즉, BUFDATA를 무한히 생산 및 소비

# Assignment 3-2:

## Solution Simulation for Bounded-buffer Problem

---

### ▶ Conditions (cont'd)

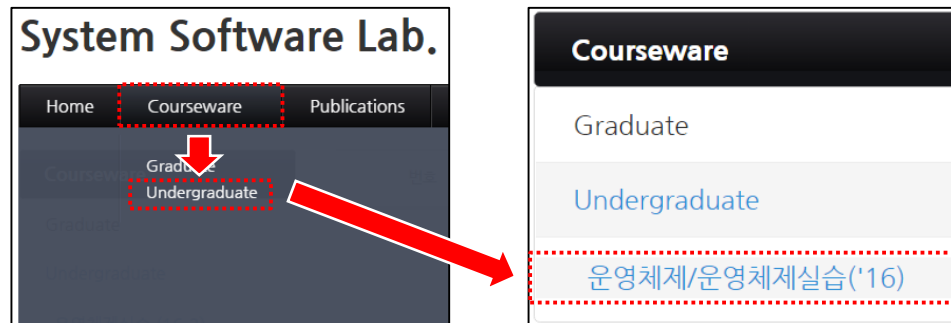
- ▶ 발생할 수 있는 모든 동기화 문제를 해결해야 함
- ▶ 프로그램 실행 시,
  - ▶ 생산자 수, 소비자 수, 최대 버퍼 크기)를 인자로 주어야 함
  - ▶ **출력 형식이 다를 경우 감점**
  - ▶ 실행 예제

```
sslslab@oslab:~$ ./bbp 3 4 5
[Producer 1392] produced 1 (buffer state: 1/5)
[Producer 1393] produced 2 (buffer state: 2/5)
[Consumer 1394] consumed 1, produced from 1392 (buffer state: 1/5)
...
```

# 과제 질문 시 Q&A 게시판 사용

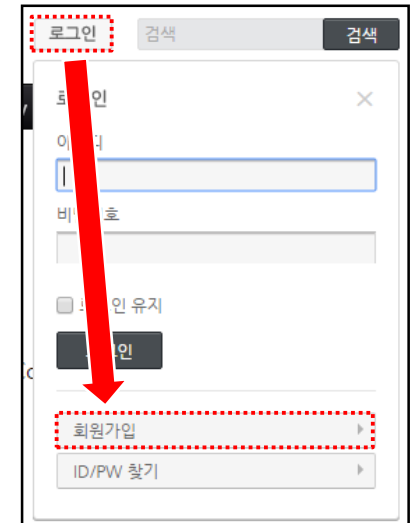
## ▶ 과제 질문은 **모두 아래의 홈페이지에서만** 받음

▶ [sswlab.kw.ac.kr](http://sswlab.kw.ac.kr) → Courseware → Undergraduate → "운영체제/운영체제실습('16)"



## ▶ 게시판을 사용하기 위해서 홈페이지 가입이 필요

- ▶ 화면 오른쪽 상단의 "로그인" → "회원가입"
- ▶ 아래의 조건이 충족 되지 않을 경우,  
**질문에 답변이 거부되거나 질문 글이 삭제될 수 있음**
  - ▶ "닉네임"이 본인 실명이 아닌 경우
  - ▶ 기타 회원 가입 시 입력한 내용이 올바르지 않은 경우



# 과제 질문 시 Q&A 게시판 사용

• 과제 질문은 **모든 아래의 홈페이지에서만** 받음

과제 마감 당일(금요일)은  
**오후 5시 이전에 올라온 질문에** 한하여 답변  
(‘2’에도 동일하게 적용)

홈페이지에 **문제가 있어 질문이 불가능한 경우,**  
담당 조교에게 직접 연락  
→ 김규식 조교, kks@kw.ac.kr

• 아래의 조건이 충족 되지 않을 경우

질문 시, 본인이 질문하는 과제를 명시할 것  
(ex. 이론 수업의 3차 과제, 1번 ...)