

2024년 2학기 운영체제실습

Assignment 4

System Software Laboratory

School of Computer and Information Engineering

Kwangwoon Univ.

Contents

- **Assignment4-1**
 - VMA information modules
- **Assignment4-2**
 - Page replacement simulation

Assignment 4-1

- 프로세스의 가상 메모리에는 파일에서 불러온 여러 정보가 적재되어 있다.
e.g. shared library, binary file...
- PID를 바탕으로 아래와 같은 **프로세스 정보를 출력**하는 Module 작성
 - 프로세스의 이름과 PID
 - 정보가 위치하는 가상 메모리 주소
 - 프로세스의 데이터 주소, 코드 주소, 힙 주소
 - 정보의 원본 파일의 전체 경로
- **Requirements**
 - 2차 과제에서 사용한 ftrace 시스템 콜(336번)에 **wrapping** 할 시스템 콜 작성
 - **Hooking 함수 명 : file_varea**
- **Hints**
 - 관련 강의자료를 참고하여 struct task_struct 부터 분석
 - 관련 강의자료의 예시 코드를 참고
 - pid_task()

Assignment 4-1

Examples

```
sslab@oslab:~$ sudo insmod file_varea.ko
sslab@oslab:~$ gcc test.c
sslab@oslab:~$ ./a.out
sslab@oslab:~$ dmesg
```

```
...
[ 6051.777952] ##### Loaded files of a process 'a.out(5482)' in VM #####
[ 6051.777954] mem(400000~401000) code(400000~40074c) data(600e10~601040) heap(810000~810000) /home/sslabs/assign4/a.out
[ 6051.777955] mem(600000~601000) code(400000~40074c) data(600e10~601040) heap(810000~810000) /home/sslabs/assign4/a.out
[ 6051.777956] mem(601000~602000) code(400000~40074c) data(600e10~601040) heap(810000~810000) /home/sslabs/assign4/a.out
[ 6051.777957] mem(7f145e379000~7f145e539000) code(400000~40074c) data(600e10~601040) heap(810000~810000) /lib/x86_64-linux-gnu/libc-2.23.so
[ 6051.777957] mem(7f145e739000~7f145e73d000) code(400000~40074c) data(600e10~601040) heap(810000~810000) /lib/x86_64-linux-gnu/libc-2.23.so
[ 6051.777958] mem(7f145e73d000~7f145e73f000) code(400000~40074c) data(600e10~601040) heap(810000~810000) /lib/x86_64-linux-gnu/libc-2.23.so
[ 6051.777959] mem(7f145e743000~7f145e769000) code(400000~40074c) data(600e10~601040) heap(810000~810000) /lib/x86_64-linux-gnu/ld-2.23.so
[ 6051.777960] mem(7f145e968000~7f145e969000) code(400000~40074c) data(600e10~601040) heap(810000~810000) /lib/x86_64-linux-gnu/ld-2.23.so
[ 6051.777960] mem(7f145e969000~7f145e96a000) code(400000~40074c) data(600e10~601040) heap(810000~810000) /lib/x86_64-linux-gnu/ld-2.23.so
[ 6051.777961] #####
```

```
sslab@oslab:~$
```

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

#define __NR_os_ftrace 336

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

test.c

Assignment 4-2

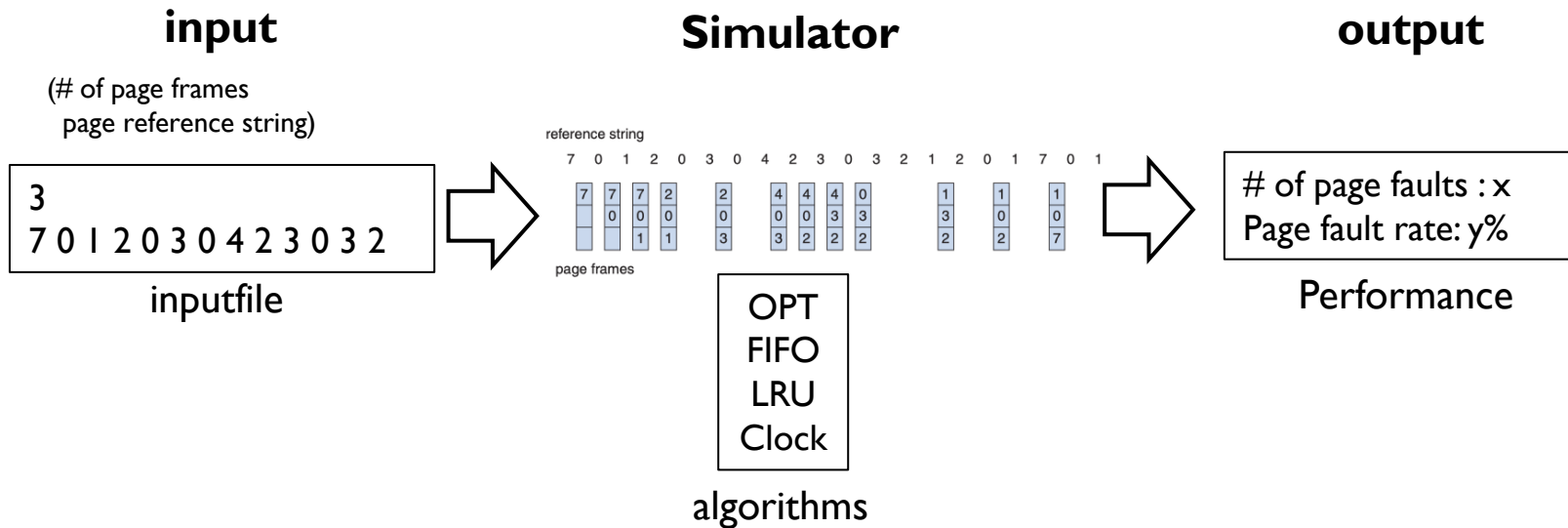
- **Simulate four page replacement algorithms:**
 - **Optimal, FIFO (First In, First Out), LRU (Least Recently Used), and Clock.**
- **Purpose of the simulation**
 - Observe and compare the performance of each algorithm
 - Number of **page faults** and the **page fault rate**.
- **Program will read the followings from an input file.**
 - Number of available page frames
 - Page reference strings

Project Description

- **Implement four page replacement algorithms.**
 - **Optimal (OPT):** Replaces the page that will not be used for the longest period in the future.
 - **FIFO (First In, First Out):** Replaces the **oldest page** that was loaded into memory.
 - **LRU (Least Recently Used):** Replaces the page that has not been used for the longest period.
 - **Clock Algorithm:** Circular structure with a reference bit to determine which pages to replace, giving pages a "second chance" before replacing them.

Project Description

- Page replacement simulator overview

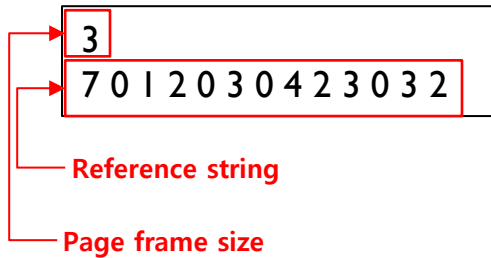


Project requirements

- **1. Implement four page replacement algorithms:**
 - Optimal, FIFO, LRU, and Clock Algorithm.
- **2. Input Handling**
 - Read the number of frames and the page reference string from an input file.
- **3. Simulate the page replacement process and track the number of page faults.**
 - Page frames are assumed to be initially empty.
- **4. Print the number of page faults and page fault rate for each algorithm.**
- **Note: If you use a static array to implement the number of page frames, assume the maximum is 1,000.**

Input

- **Page reference string**
 - Read from an input file.
 - Input file contains two parts:
 - **Number of page frames:** An integer representing the number of available frames.
 - **Page reference string:** Integers represent the pages requested by the system in order.
- Input file example (input.1)



Simulator

- **Command-line Usage**
 - **“page_replacement_simulator inputfile”**
 - Read the number of frames and page reference string from the inputfile
 - Simulate the four algorithms (Optimal, FIFO, LRU, and Clock).
 - Examples
 - **Page_replacement_simulator input.1**
 - Simulate all page replacement algorithms using the data file "input.1".

Output

- **Sample output**

Optimal Algorithm:
Number of Page Faults: X
Page Fault Rate: Y%

FIFO Algorithm:
Number of Page Faults: X
Page Fault Rate: Y%

LRU Algorithm:
Number of Page Faults: X
Page Fault Rate: Y%

Clock Algorithm:
Number of Page Faults: X
Page Fault Rate: Y%

- 보고서에 각 알고리즘 별 성능결과를 분석 비교할 것.

Output

- Input file(input.1)

```
os2024123456@ubuntu:~/assgin4/4-2$ cat input.1
3
7 0 1 2 0 3 0 4 2 3 0 3 2
```

- \$/page_replacement_simulator input.1

```
os2024123456@ubuntu:~/assgin4/4-2$ ./page_replacement_simulator input.1
Optimal Algorithm:
Number of Page Faults: 7
Page Fault Rate: 53.85%

FIFO Algorithm:
Number of Page Faults: 10
Page Fault Rate: 76.92%

LRU Algorithm:
Number of Page Faults: 9
Page Fault Rate: 69.23%

Clock Algorithm:
Number of Page Faults: 8
Page Fault Rate: 61.54%
```

Report Requirements

- **Ubuntu 20.04.6 Desktop 64bits 환경에서 채점**
- **Copy 발견 시 0점 처리**
- **보고서 구성**
 - **보고서 표지**
 - 수업 명, 과제 이름, 담당 교수님, 학번, 이름 필히 명시
 - 과제 이름 → Assignment #4
 - **과제 내용**
 - Introduction
 - 과제 소개 - 4줄 이상(background 제외) 작성
 - Result
 - 수행한 내용을 캡처 및 설명
 - 고찰
 - 과제를 수행하면서 느낀 점 작성
 - Reference
 - 과제를 수행하면서 참고한 내용을 구체적으로 기록
 - 강의자료만 이용한 경우 생략 가능

Report Requirements

- **Source**
 - Assignment 4-1(2개)
 - file_varea.c
 - Makefile
 - Assignment 4-2(2개)
 - page_replacement_simulator.c
 - Makefile
- 각 코드 파일은 디렉토리를 나누어서 제출
 - E.g)
 - Assignment4-1/ file_varea.c
/ Makefile
 - Assignment4-2/ page_replacement_simulator.c
/ Makefile
- **Copy 발견 시 0점 처리**

Report Requirements

- Softcopy Upload

- 제출 파일
 - 보고서 + 소스파일 [하나의 압축 파일로 압축하여 제출(tar.xz)]
 - 보고서(.pdf. 파일 변환)
 - 소스코드(**Comment 반드시 포함**)

- 보고서 및 압축 파일 명 양식
- **OS_Assignment4_수강분류코드_학번_이름** 으로 작성

수강요일	이론1 월6수5	이론2 목3	실습 금56
수강분류코드	A	B	C

- 예시 #1)-**이론(월6수5)만** 수강하는 학생인 경우
 - 보고서 OS_Assignment1_**A**_2024123456_홍길동.pdf
 - 압축 파일 명: OS_Assignment1_**A**_2024123456_홍길동.tar.xz
- 예시 #2)-**이론(월6수5 or 목3)과 실습** 모두 수강하는 학생인 경우
 - 보고서 OS_Assignment1_**C**_2024123456_홍길동.pdf
 - 압축 파일 명: OS_Assignment1_**C**_2024123456_홍길동.tar.xz
 - + **"해당 이론반 txt 파일 제출"**

Report Requirements

- 실습 수업을 수강하는 학생인 경우

- 실습 과목에 과제를 제출(.tar.xz)
- 이론 과목에 간단한 .txt 파일로 제출

📄 실습수업때제출했습니다.

2022-08-29 오후 3:58

텍스트 문서

0KB

- 이론 과목에 .txt 파일 미 제출 시 감점
- .tar.xz 파일로 제출 하지 않을 시 감점

- 과제 제출

- KLAS – 강의 과제 제출
- 2024년 11월 21일 목요일 23:59까지 제출
- 딜레이 받지 않음
 - 제출 마감 시간 내 미제출시 해당 과제 0점 처리(예외 없음)