

[506489] 시스템프로그래밍

실습 #13 문제 및 보고서

이름	곽영주
학번	20175105
소속 학과/대학	빅데이터
분반	01 (담당교수: 김태운)

<주의사항>

- 개별 과제입니다. (팀으로 진행하는 과제가 아니며, 모든 학생이 보고서를 제출해야 함)
- 각각의 문제 바로 아래에 답을 작성 후 제출해 주세요.

- 소스코드/스크립트 등을 작성 한 경우, 해당 파일의 이름도 적어주세요.

- SmartLEAD 제출 테드라인:

- 월요일 분반: 다음 실습 시간 전날까지 (일요일 까지)

- 수요일 분반: 다음 실습 시간 전날까지 (화요일 까지)

- 테드라인을 지나서 제출하면 24 시간 단위로 20%씩 감점 (5 일 경과 시 0 점)

- 주말/휴일/학교행사 등 모든 날짜 카운트 함

- 부정행위 적발 시, 원본(보여준 사람)과 복사본(베낀 사람) 모두 0 점 처리함

- 예외 없음

- SmartLEAD 에 아래의 파일을 제출해 주세요

- 보고서 (PDF 파일로 변환 후 제출 권장하나, WORD 형식으로 제출도 가능)

- 보고서 파일명에 이름과 학번을 입력해 주세요.

- 소스코드, 스크립트, Makefile 등을 작성해야 하는 경우, 모든 파일 제출 (zip 파일로 압축하여 하나의 파일로 제출)

<개요>

이번 과제는 메모리 매핑에 관한 내용입니다.

<실습 과제>

[Q 0] 요약 [배점 10]

이번 과제에서 배운 내용 또는 과제 완성을 위해서 무엇을 했는지 2~3 문장으로 요약하세요.

답변: 이번 과제를 통해 메모리 매핑에 대한 개념을 잡을 수 있었습니다. 또한, 파일 입출력 대신 훨씬 편하고 속도가 빠른 것 같다.

[Q 1] Ex4 따라하기 [배점 20]

강의자료에 있는 Ex4 와 동일한 프로그램을 작성하고 실행하세요. 터미널 출력 결과가 강의자료에 있는 결과와 동일해야 합니다.

터미널 출력 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w13/q01$ cat mmap.dat
HALLYM
BOOK
yeongju@vm-ubuntu20:~/sp2021-2/w13/q01$ ./ex4 mmap.dat
HALLYM
BOOK
-----
DALLYM
BOOK
yeongju@vm-ubuntu20:~/sp2021-2/w13/q01$ cat mmap.dat
DALLYM
BOOK
yeongju@vm-ubuntu20:~/sp2021-2/w13/q01$
```

[Q 2] Ex5 따라하기 [배점 20]

강의자료에 있는 Ex5 와 동일한 프로그램을 작성하고 실행하세요. 터미널 출력 결과가 강의자료에 있는 결과와 동일해야 합니다.

터미널 출력 화면을 캡처하고 본 문서에 첨부하세요. 소스코드도 제출해야 합니다.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w13/q02$ cat mmap.dat
HALLYM BOOK
yeongju@vm-ubuntu20:~/sp2021-2/w13/q02$ ./ex5 mmap.dat
1. Parent Process : addr=HALLYM BOOK
1. Child Process : addr=HALLYM BOOK
2. Child Process : addr=xALLYM BOOK
2. Parent Process : addr=xALLYM BOOK
3. Parent Process : addr=xyLLYM BOOK
3. Child Process : addr=xyLLYM BOOK
yeongju@vm-ubuntu20:~/sp2021-2/w13/q02$ cat mmap.dat
xyLLYM BOOK
yeongju@vm-ubuntu20:~/sp2021-2/w13/q02$
```

[Q 3] 메모리 매핑을 이용한 게임 만들기 1 [배점 30]

부모-자식 프로세스가 메모리 매핑을 이용해서 간단한 게임을 하는 프로그램 `memPlay.c` 를 작성하세요. 먼저, 게임판의 초기 상태를 저장할 `game.txt` 파일을 만드세요. 파일에는 50 개의 'p' 문자와 50 개의 'c' 문자가 순차적으로 저장되어 있습니다. 파일을 열면, 처음 50 개 문자는 'p'이고, 그 다음 50 개 문자는 'c' 입니다. 부모 프로세스와 자식 프로세스는 메모리 매핑을 통해 `game.txt` 파일을 `addr` 변수에 매핑합니다 (매핑할 파일 이름은 명령행 인자로 전달합니다). 매핑 후, 부모 프로세스와 자식 프로세스는 각각 100 번 동안 아래의 동작을 반복합니다.

<게임 알고리즘: 100 번 반복>

[Q 4] 메모리 매핑을 이용한 게임 만들기 2 [배점 20]

위의 [Q 5]에서 작성한 프로그램을 수정해서 `memPlay2.c` 프로그램을 작성하세요. [Q 5]와 같이 부모-자식 프로세스간 게임을 진행합니다. 게임은 아래와 같이 진행됩니다. 부모 프로세스와 자식 프로세스는 각각 100 번 동안 아래의 동작을 반복합니다.

- 1) 부모: 0~99 에서 하나의 정수 `k` 를 무작위로 선택.
 자식: 알고리즘에 따라 0~99 에서 하나의 정수 `k` 를 선택
- 2) 부모는 `addr[k]`를 ‘p’로 바꾸고, 자식은 ‘c’로 바꿈
- 3) 1 초간 sleep

자식 프로세스가 k 를 선택하는 알고리즘:

부모는 무작위로 `k` 를 선택하지만, 자식 프로세스는 ‘(거의 무조건) 지지않는’ 알고리즘을 가지고 있으며, 알고리즘에 따라서 `k` 를 선택합니다. 자식 프로세스가 지지 않도록 `k` 를 선택하는 알고리즘(=k-선택-알고리즘)을 제안하고 구현하세요. 자식 프로세스는 부모가 선택한 `k` 값을 알 수 없습니다. 자식 프로세스는 한번에 하나의 값만 ‘c’로 설정할 수 있습니다.

- 자식 프로세스는 매 iteration 마다 `k` 를 한 번만 선택할 수 있습니다.
- 부모 프로세스가 어떤 `k` 값을 선택했는지 알아낼 수 없고, 알아내어도 안됩니다.

[Task] 아래와 같은 순서로 프로그램을 실행하세요.

(먼저, [Q 5]와 같은 방식으로 `game.txt` 파일에 ‘p’ 문자와 ‘c’ 문자를 각각 50 개씩 입력하고 저장하세요)

```
$cat game.txt // 파일에 저장된 내용 출력
```

```
$/memPlay2 game.txt // 게임 진행
```

(게임이 종료되면 아래의 명령을 입력하세요)

```
$/countP game.txt // 게임 결과 확인
```

