

## 2015 년 2 학기 운영체제 과제 #5

2015/11/20 (금)

### 1. 과제 내용

메모리 맵 파일을 이용하여 프로세스간 자료를 공유하고, 자료가 변경되면 유닉스 시그널을 이용해 다른 프로세스에게 알려주는 프로그램을 구현하시오.

### 2. 요구 사항.

- A. C 언어와 기본 C 라이브러리(glibc)만 이용하고, 소스 파일 `person.c` 한 개에 모든 기능을 구현하며, 실행 파일 이름은 `person` 입니다. 실행 파일은 다음과 같은 옵션을 제공합니다.

```
$ ./person
usage: ./person [-f file] [-w] [-s value] attr_name
  -f file   : 공유할 파일 이름, 지정하지 않을 경우 './person.dat'
  -w        : 변경 사항 감시 모드로 동작
  -s value  : 지정한 'attr_name' 에 대한 값(value) 변경, 지정하지 않을 경우 값 출력
```

- B. 메모리 맵 파일을 이용해 파일에 맵핑된 자료구조 값을 변경하거나 가져와서 출력합니다.
- 맵핑할 자료구조는 다음과 같이 정의됩니다. (샘플 소스 `person.h` 파일 참고)

```
typedef struct
{
    pid_t watchers[NOTIFY_MAX];

    char  name[64];
    int   age;
    int   gender; /* 1:male, 0:female */
    char  phone[24];
    char  homepage[80];
    char  twitter[80];
    char  facebook[80];
} Person;
```

- '-s' 옵션을 지정하지 않았을 경우 인수로 지정한 속성 이름(attr\_name)에 대한 값을 출력하고 종료합니다. 예를 들면 다음과 같습니다.

```
$ ./person name
Lee Oisoo
$ ./person age
66
$ ./person gender
1
$ ./person twitter
http://twitter.com/oisoo
$ ./person cyworld
invalid attr name 'cyworld'
```

- 구조체 필드를 소스 레벨에서 직접 접근하면 안됩니다.

(예: `person->age`, `person->twitter`)

- '-s' 옵션을 지정했을 경우 인수로 지정한 속성 이름(attr\_name)의 값을 변경하고 변경된 결과값을 출력합니다. 예를 들면 다음과 같습니다.

```
$ ./person -s "Sunjin Yang" name
Sunjin Yang
$ ./person -s 41 age
41
```

- 출력할 때와 마찬가지로 정수형과 문자열을 구분해야 하고, 구조체 필드를 소스 레벨에서 직접 접근하면 안됩니다.
- 문자열을 변경할 때 배열 크기 제한으로 인한 오버플로우(overflow) 검사는 안해도 됩니다.

- C. 속성을 변경하면 대기 중인 프로세스에게 변경된 속성을 알려주고, 대기 중인 프로세스는 신호를 받으면 변경된 속성과 시그널을 전송한 프로세스 ID(pid)를 출력합니다.

- '-w' 옵션을 지정하면 대기 모드(watch mode)로 동작합니다. 예를 들어, 위의 예처럼 다른 프로세스가 속성을 변경하면 대기 중인 프로세스는 다음과 같이 출력합니다.

```
$ ./person -w
watching...
name: 'Sunjin Yang' from '7947'
age: '41' from '7982'
```

### 3. 구현 방법

- ✓ 프로그램은 실행하자마자 인수로 지정한 파일을 열고, `mmap()` 함수를 이용해 파일의 시작 위치부터 `Person` 구조체 크기만큼 프로세스 주소 공간에 맵핑한 뒤 그 시작 주소를 '`Person *`' 타입의 포인터 변수에 할당합니다. 파일이 없을 경우 `Person` 구조체 크기만큼 새 파일을 생성하고 모든 값을 `0x00` 으로 채운 다음 맵핑합니다.
- ✓ 속성 위치에 대한 이름을 얻으려면 `person_lookup_attr_with_attr()` 함수를 이용합니다. 속성 이름에 대한 구조체 내에서의 필드(field) 위치는 `person_get_offset_of_attr()` 함수를 이용합니다. (person.h 샘플 소스 참고)
- ✓ 구조체 주소에 필드 위치값(offset)을 더한 주소가 가리키는 내용을 정수형 / 문자열에 따라 다르게 출력합니다. 속성이 문자열인지 정수형인지 여부는 제공되는 샘플 소스에 있는 `person_attr_is_integer()` 함수를 이용합니다.

- ✓ 구조체 필드 값을 읽고 쓸 때 포인터 변수 연산을 이용해야 합니다. 예를 들어 정수형 포인터로 변환해 값을 쓸 경우 `*(int *)ptr = int_value` 처럼 작성하는 기법이 필요합니다.
- ✓ 대기 모드로 들어가기 전에 해당 파일을 맵핑한 `Person` 자료 구조의 `watchers[]` 배열 중에서 0 인 항목을 찾아 자신의 프로세스 ID(pid)를 채우고, `SIGUSR1 / SIGINT / SIGTERM` 시그널 핸들러를 등록합니다. 0 이 아닌 항목이 없을 경우 무조건 배열 첫 번째 항목을 덮어씁니다. 대기 모드가 아닌 다른 프로세스는 '-s' 옵션을 지정해서 속성을 변경했을 때 `watchers[]` 배열에서 0 이 아닌 모든 프로세스 ID 에게 해당 속성이 변경되었을 알리는 `SIGUSR1` 시그널을 전송합니다.
- ✓ `SIGUSR1` 시그널은 다른 프로세스가 속성을 변경했을 때 속성의 필드 위치 값(offset)을 전달하는데 사용하고, `SIGINT / SIGTERM` 시그널은 프로그램을 종료할 때 `watchers[]` 배열에 등록한 자신의 프로세스 ID 를 0 으로 만들고 프로그램을 종료하는데 사용합니다. 다른 프로세스는 '-s' 옵션을 지정해서 속성을 변경한 다음 `watchers[]` 배열에서 0 이 아닌 모든 프로세스 ID 에게 해당 속성이 변경되었을 알리는 `SIGUSR1` 시그널을 전송합니다.
- ✓ 시그널 전송은 `sigqueue()` 함수를 이용하여 속성의 필드 위치 값(offset)까지 전달합니다. 시그널 전송에 실패하면 `watchers[]` 배열에서 해당 프로세스 ID 항목을 0 으로 초기화합니다.  
`sigqueue(3)` 함수는 POSIX.1-2001 표준입니다. 인수를 전달하는 구체적인 방법은 매뉴얼 페이지를 참고하기 바랍니다.
- ✓ 시그널 수신은 `sigaction()` 함수를 `SA_SIGINFO` 플래그와 함께 이용하고 `siginfo_t` 자료구조 안의 `si_pid, si_value` 필드를 이용하면 됩니다.
- ✓ 샘플 소스에는 기본적인 명령어 인수 파싱 루틴이 포함되어 있지만, 이를 무시하고 직접 만들어도 상관없습니다.

#### 4. 채점 기준

- ✓ 보고서 (20 점) : 아래 목차를 따라 작성합니다.
  1. 소개
  2. 관련 연구 (메모리 맵 파일 / 시그널 관련 조사 포함) (5)
  3. 문제 해결 방법 (5)
  4. 실행 결과 (3)
  5. 추가 및 개선할 부분과 결론 (5)

- 6. 참고 자료 / 문헌 (2)
- 7. 소스 코드 (없을 경우 보고서 점수 0 점 처리)
- ✓ 구현 (30 점)
  - 요구사항 A 준수 및 소스 코드 가독성(readability)과 주석(comments) 여부 (5 점)
  - 요구사항 B 구현 (15 점)
  - 요구사항 C 구현 (10 점)

## 5. 제출 방법

- ✓ 마감시간: **2014/12/4 오후 6 시 (2 주)**
- ✓ 감점: 늦게 제출할 경우 1 일 10 점 감점
- ✓ 제출 방법
  - 소스 및 소프트카피: 소스파일, Makefile, README 파일 그리고 소프트 카피 포함에서 압축하고, 파일 이름은 '반\_이름\_과제번호.zip' 형식의 이름을 사용 (예: 가\_홍길동\_hw5.zip)
  - [vanillo@realtime.ssu.ac.kr](mailto:vanillo@realtime.ssu.ac.kr) 로 제출

## 6. 기타

- ✓ 과제 해결 방법에 대한 논의나 도움은 서로간에 주고 받을 수 있겠지만 과제 자체는 전적으로 본인이 직접 스스로 해야 합니다. 만일 복사본이 발견되면 복사를 한 학생과 더불어 원본 제공자 또한 해당 과제 점수를 전체 0 점 처리합니다.
- ✓ 과제 관련 궁금한 점이나 문의 사항은 수업 게시판을 이용하시기 바랍니다.

<https://groups.google.com/forum/#!forum/ssu-smyang>