

[506489] 시스템프로그래밍

실습 #14 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 **마지막** 실습 과제이며, 시그널을 다루는 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 시그널이 무엇인지 배울 수 있었습니다.

또한, 이번 과제를 완성하면서 시그널의 블로킹 및 알람, 인터벌 타이머가 어떠한 방식으로 돌아가는 배울 수 있었습니다.

[Q 1] 시그널 [18 점]

시그널 집합을 이용해 모든 시그널을 집합에 추가하고, 이 중에서 SIGBUS 가 추가되었는지 여부를 확인하는 프로그램을 작성하세요. SIGBUS 가 설정 되었다면 “SIGBUS is set.”이라는 문자열을 출력하고, 그렇지 않으면 “SIGBUS is not set.”이라는 문자열을 출력하세요.

터미널 화면을 캡처해서 본 문서에 첨부하고, 소스코드를 제출하세요.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w14/q01$ ./sigbus
Add all signal..
SIGBUS is set.
yeongju@vm-ubuntu20:~/sp2021-2/w14/q01$
```

[Q 2] 시그널 [18 점]

‘signal’ 함수를 이용해서 SIGQUIT 시그널 핸들러를 등록하고, 핸들러 호출 시 “SIGQUIT handler is called!”라는 간단한 메시지를 출력하는 프로그램을 작성하세요.

터미널 화면을 캡처해서 본 문서에 첨부하고, 소스코드를 제출하세요.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w14/q02$ ./handler
wait input [ctrl + \]..
^\\SIGQUIT handler is called!
Received Signal: Quit
yeongju@vm-ubuntu20:~/sp2021-2/w14/q02$
```

[Q 3] 시그널 [18 점]

반복문과 `sleep(1)` 함수를 사용해서 1 초 간격으로 “Hallym Software”를 총 10 번 출력하는 프로그램을 작성하세요. 프로그램이 `Ctrl+C` 로 종료되지 않도록 `sigprocmask` 함수를 사용해서 블로킹 하세요.

터미널 화면을 캡처해서 본 문서에 첨부하고, 소스코드를 제출하세요.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w14/q03$ ./sigprocmask
Hallym Software
^CHallym Software
^C^C^CHallym Software
^C^C^CHallym Software
^C^CHallym Software
^CHallym Software
^CHallym Software
^CHallym Software
^CHallym Software
Hallym Software
yeongju@vm-ubuntu20:~/sp2021-2/w14/q03$
```

[Q 4] 시그널 [18 점]

‘alarm’ 함수를 사용해서 2 초마다 SIGALRM 시그널을 생성하고, 해당 핸들러에서 “At handler : Wake up!” 메시지를 출력하는 프로그램을 작성하세요. SIGALRM 시그널의 핸들러가 실행되는 동안 SIGINT 를 제외하고 모든 시그널을 블로킹 하세요.

메인함수를 다음과 같이 코딩하세요.

```
int main(void) {  
    signal(SIGALRM, handler);  
    for( int i = 0 ; i < 5 ; i++ ) {  
        alarm(2); // 2 초 설정  
        printf("Pause ...\n");  
        pause();  
    }  
    printf("Bye~\n");  
    return 0;  
}
```

프로그램 실행 결과는 아래와 같아야 합니다.

```
Pause ...  
At handler : Wake up!  
Pause ...  
At handler : Wake up!  
Pause ...  
At handler : Wake up!  
Pause ...  
At handler : Wake up!  
Pause ...  
At handler : Wake up!  
Bye~
```

터미널 화면을 캡처해서 본 문서에 첨부하고, 소스코드를 제출하세요.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w14/q04$ ./alarm
Pause ...
At handler : Wake up!
Pause ...
At handler : Wake up!
Pause ...
At handler : Wake up!
Pause ...
At handler : Wake up!
Pause ...
At handler : Wake up!
Bye~
yeongju@vm-ubuntu20:~/sp2021-2/w14/q04$
```

[Q 5] 인터벌 타이머 [18 점]

‘setitimer’ 를 사용해서 1 초마다 SIGALRM 핸들러에서 “Nice to meet you!” 메시지를 출력하는 프로그램을 작성하세요.

터미널 화면을 캡처해서 본 문서에 첨부하고, 소스코드를 제출하세요.

답변:

```
yeongju@vm-ubuntu20:~/sp2021-2/w14/q05$ ./timer
Nice to meet you!
Nice to meet you!
Nice to meet you!
Nice to meet you!
Nice to meet you!
Nice to meet you!
Nice to meet you!
^C
yeongju@vm-ubuntu20:~/sp2021-2/w14/q05$
```

끝! 수고하셨습니다 ☺