

Operating Systems Project01

-Implementing simple system call and interrupt on xv6

Due date : 11:59 pm April 4

Originated by Scalable Computing Systems Lab.,
Revised by Software Platform Lab.

Project01

- 운영체제 실습 첫 과제의 목표는 다음과 같습니다.
 1. Xv6 설치 및 빌드 확인
 2. Cscope 활용한 코드 분석
 3. 시스템 콜 및 인터럽트 구현
 4. GitLab을 통한 과제 제출 방법 숙지
- 이번 과제는 앞으로의 본격적인 수업 및 과제에 앞서 워밍업을 하는 과제입니다. 편안한 마음으로 과제에 임해주시면 감사하겠습니다.

Project01-1

- 부모프로세서의 id를 가져오는 **getppid()** 시스템 콜을 구현하고, 이를 확인하기 위한 유저프로그램을 작성한다.
- 유저프로그램의 이름은 project01_1로 작성한다.

Tips:

- It is helpful to refer to the **getpid()** system call
- Use cscope to analyze getpid()

Project01-1

- 유저 프로그램 실행예시

```
xv6...  
cpu0: starting  
sb: size 1000 nblocks 941 ninodes 200 nlog 30 logstart 2  
inodestart 32 bmap start 58  
init: starting sh  
$ project01_1  
My pid is 3  
My ppid is 2
```

Project01-2

- There are 256 interrupt descriptors in xv6. Only interrupt number 64 (T_SYSCALL) is the one switching entrypoint from user to kernel.
- Let's make another entrypoint with interrupt number 128. This interrupt just print "user interrupt (trap number) called!" and exit the process.
- Implement this user application that invokes "int 128". **NOTE !!** result should not invoke the error exception (interrupt 13).

```
// user application source code
...
1 int
2 main(int argc, char *argv[])
3 {
4     __asm__("int $128");
5     return 0;
6 }
```

```
// xv6...
$ project01_2
user interrupt 128 called!
$
```

```
$ project01_2
pid 4 user_app: trap 13 err 1026 on cpu 0
eip 0x3 addr 0x0--kill proc
$
```

평가

- **Completeness(60)**: 명세의 요구조건에 맞게 xv6가 올바르게 동작해야 합니다. (xv6가 빌드되지 않으면 채점 불가)
- **Wiki&Comment(40)**: 유저 프로그램과 위키를 기준으로 채점이 진행되므로 위키는 최대한 상세히 작성되어야 합니다.
- **Deadline**: 데드라인을 반드시 지켜야 하며, 데드라인 이전 마지막으로 commit/push된 코드를 기준으로 채점됩니다.
- **Do NOT copy or share your code.**

Wiki

- Wiki에는 다음의 항목들이 서술되어야 합니다.
 - **디자인**: 구현해야하는 시스템 콜과 인터럽트를 어떻게 구현해야 하는지, 추가/변경되는 컴포넌트들이 어떻게 상호작용하는지 등을 서술합니다.
 - **구현**: 실제 구현과정에서 변경하게 되는 코드영역에 대하여 서술합니다.
 - **실행 결과**: 유저프로그램이 올바르게 동작하고 있음을 확인할 수 있는 실행 결과와 이에 대한 설명을 서술합니다.
 - **트러블슈팅**: 코드 작성 중 생겼던 문제와 이에 대한 해결 과정을 서술합니다.

주의사항

- Xv6 소스코드는 gitlab을 통해 제출합니다.
- Wiki는 .doc 형식으로 LMS 과제 란을 통해 제출합니다.
- Wiki 파일명은 OS_Project01_(학번).doc 으로 작성합니다.
- Due date : 11:59 pm April 4 이후 제출시 0점 처리

Thank you.
