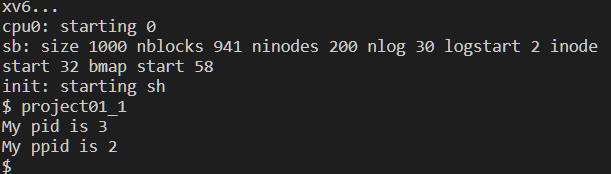
**OS\_Project01\_2017069598 박상지**

1. project01-1

* 디자인: “project01-1.c 파일을 만들어서 “getpid()” 함수와 “getppid()” 함수를 호출시켰고, getppid()함수는 getpid() 함수가 존재하는 sysproc.c에 구현하면 새로운 c파일 만들지 않고 후에도 깔끔하게 코드를 추가하여 구현을 하였습니다.
* 구현:

1. ‘sysproc.c’ 파일 안에 ‘getppid()’를 함수를 94line에 정의를 하였다. ‘getppid()’ 함수는 부모프로세서의 ‘pid’를 호출하기 위해 인자 값이 없이 int형으로 부모프로세서의 id값을 반환시킨다. 동일 파일에 100line에서 ‘sys\_getppid()’ wrapper 함수를 선언을 하였다. 인자를 받는 게 필요 없기 때문에 반환 값은 단순하게 실행시킬 ‘getppid()’함수를 호출시켰다.
2. wrapper 함수에 대한 선언이 필요하기 때문에 ‘syscall.h’과‘syscall.c’에 sys\_getppid()에 대한 정의를 해줬다.
3. 유저에서도 시스템콜을 할 수 있게 ‘user.h’에 ‘sys\_getppid()’에 대한 정의를 해줬다.
4. 마지막으로 ‘usys.S’에 ‘getppid()’함수를 정의를 해줬다.

* 실행 결과:



현재 실행된 프로세서 getpid()에서는 id값 3이 출력되며, getppid()에서는 부모 프로세서의 id값인 2이 출력된다. 현재 project01\_1을 한번 실행시켜서 결과를 이미지를 보여줬지만 여러 번 실행할 때 마다 부모의 프로세서는 유지되지만 현재 프로세서 id값은 1씩 계속 증가한다. id값은 전의 값이 아닌 최종적으로 사용한 값 다음의 비어 있는 id 값을 할당 받는다. 그러므로 해당 코드는 현재 프로세서 id 값이 증가하는 것으로 보아 올바르게 동작하고 있다는 것을 예상할 수 있다.

* 트러블 슈팅:

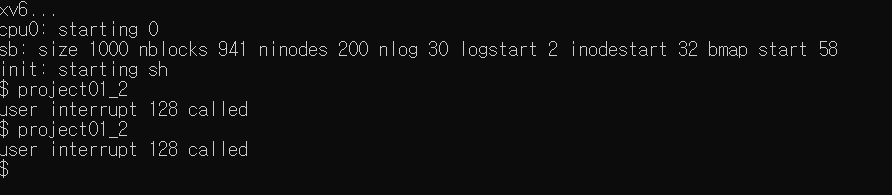
‘project01-1’을 수행할 때 실습 동영상 lab3를 참고를 하여 전체적으로 코드를 구현하는 부분에 있어서 어려움이 없었다. 다만 ‘Makefile’에서 project01-1.c 파일을 정의를 안 하여 make할 때 오류가 났었지만 해당 ‘Makefile’에서 오류인 것을 인지 못하고 강의영상을 다시 반복 시청하여 실수를 알아챌 수 있었다.

1. project01-2

* 디자인: ‘project01-2’에서는 단순히 인터럽트 128에서 인터럽트 오류 없이 인터럽트 발생 출력만 하면 된다. 인터럽트 128만 유저모드에서 호출하고 실행시킬 수 있게 설정하였다. trap frame의 trap number가 128이면 프로젝트 명세서에 나온 예시처럼 trap number와 함께 출력하고 프로세서를 종료시켰다.
* 구현:

1. trap.c에서 인터럽트 128에 대한 예외 처리를 해주기 위해 ‘tvinit()’ 함수에서 인터럽트를 초기화를 시켜주는 코드에서 인덱스 값이 128일 때 해당 인터럽트를 ‘DPL\_USER’로 설정하여 유저모드에서 호출하고 실행시킬 수 있게 작성하였다.
2. 실제로 인터럽트 128이 발생하면 ‘trap()’함수에서 switch문제 128 인터럽트에 대한 조건을 추가하였다. 명세서에 맞게 출력문을 작성하였고, 출력하는 동시에 ‘exit()’을 사용하여 프로세서를 종료시켰다.

* 실행 결과:



출력된 결과를 이미지를 추가하였다. 인터럽트 128이 발생하여도 인터럽트 경고 메시지메 없는 것을 확인할 수 있다. 그 이유는 ‘trap()’ 함수에 있는 조건문에서 출력 동시에 ‘exit()’함수를 사용했기 때문에 에러 없이 인터럽트 128에 대한 유저모드에서 호출된 것을 확인되었다. 그러므로 해당 코드는 올바르게 작동하는 것으로 예상된다.

* 트러블 슈팅:

해당 프로젝트도 실습 lab04를 참고하여 어렵지 않게 코드를 작성할 수 있었다. 다만 trap()함수에서 exit()를 선언하지 않아 명세서에 적힌 오류 사례처럼 결과가 나왔다. 명세서를 잘 읽었으면 exit the process에서 힌트를 얻어 해당 코드를 마무리 지었다.