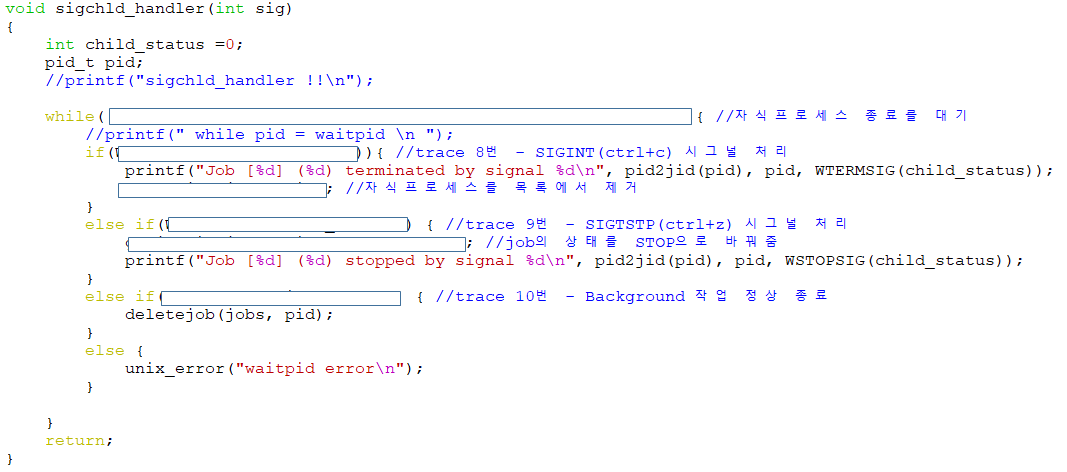
Shell Lab – 3 구현 참고차료

2018년 11월 22일

▣ 참고 : <http://codetravel.tistory.com/42>

* sigchld\_handler 구현



1. **먼저 while문 내의 ‘자식 프로세스 종료를 대기’ 조건을 구현해야 함.**

( pid = waitpid( , , , ) ) > 0 의 형식임.

waitpid() 함수는 wait 함수보다 좀 더 정교하게 자식 프로세스의 변화에 따른 제어를 할 수 있다.  
특정 프로세스가 종료되기를 기다리도록 지정할 수 있고, wait() 함수와 다르게 부모 프로세스와 자식 프로세스 모두 각자 할 일을 할 수 있도록 설정할 수 있다.   
  


pid\_t waitpid(pid\_t **pid**, int\* **status**, int **options**);

1. **pid** : 자식 프로세스의 id.  
   만약 pid < -1 이면 pid의 절대값과 동일한 프로세스 그룹 id의 모든 자식 프로세스의 종료를 기다린다.  
   만약 pid == -1 이면 모든 자식 프로세스의 종료를 기다린다. 이 때에 waitpid()는 wait()과 동일한 기능을 하게 된다.  
   만약 pid == 0 이면 현재 프로세스의 프로세스 그룹 id와 같은 프로세스 그룹 id를 가지는 모든 자식 프로세스의 종료를 기다린다.  
   만약 pid > 0 이면 pid 값에 해당하는 프로세스의 종료를 기다린다.
2. **status** : wait() 함수와 같은 역할. 필요 없으면 NULL.
3. **options** : 다음의 옵션 값들을 OR 연산으로 사용할 수 있다

- WNOHANG : 기다리는 PID가 종료되지 않아서 즉시 종료 상태를 회수 할 수 없는 상황 에서 호출자는 차단되지 않고 반환값으로 0을 받음  
    - 0 : wait() 함수와 같이 자식 프로세스의 종료를 기다림.  
    - WUNTRACED : 중단된 자식 프로세스의 상태를 받음  
    - WCONTINUED : 중단 되었다가 재개된 자식 프로세스의 상태를 받음

* **hint** : 모든 자식 프로세스의 종료를 기다리며, option은 SIGINT의 경우와 SIGSTOP의 경우를 잘 생각해 볼 것. 위 3개의 인자 값이 무엇을 써야 될지를 고민해 볼 것. 2번째 인자는 포인터의 변수인 것을 고려할 것.

1. **if 문에는 status의 상태를 확인한다. 본 과제의 경우 SIGNT, SIGSTOP, 정상종료 3가지로 나뉨. 3가지의 경우를 각각 조건문으로 나누어 구현한다. 아래의 매크로 함수를 참고하여 사용할 것.**

status 는 프로세스의 상태를 가져오기 위해서 사용한다. status 가 NULL 이 아닐경우 status 가 가리키는 위치에 프로세스의 상태정보를 저장한다. 다음의 메크로들을 통해서 상태정보를 알아가져올수 있다.

[**WIFEXITED**](https://www.joinc.co.kr/modules/moniwiki/wiki.php/manSearch?google=none&name=WIFEXITED)**(status)**

자식이 정상적으로 종료되었다면 non-zero 이다.

[**WEXITSTATUS**](https://www.joinc.co.kr/modules/moniwiki/wiki.php/manSearch?google=none&name=WEXITSTATUS)**(status)**

[exit](https://www.joinc.co.kr/modules/moniwiki/wiki.php/manSearch?google=none&name=exit)()를 호출하기 위한 인자나 return 값이 설정되고 종료된 자식의 반환 코드의 최하위 8비트를 평가한다. 이 매크로는 정상정료 - [WIFEXITED](https://www.joinc.co.kr/modules/moniwiki/wiki.php/manSearch?google=none&name=WIFEXITED)(status) - 일때만 평가된다.

[**WIFSIGNALED**](https://www.joinc.co.kr/modules/moniwiki/wiki.php/manSearch?google=none&name=WIFSIGNALED)**(status)**

자식프로세스가 어떤 신호때문에 종료되었다면 참을 반환한다.

[**WTERMSIG**](https://www.joinc.co.kr/modules/moniwiki/wiki.php/manSearch?google=none&name=WTERMSIG)**(status)**

자식프로세스를 종료하도록한 신호의 번호를 반환한다. 당연히 WIFSIGNALED 가 non\_zero 일 경우에만 사용할수 있다.

[**WIFSTOPPED**](https://www.joinc.co.kr/modules/moniwiki/wiki.php/manSearch?google=none&name=WIFSTOPPED)**(status)**

반환의 원인이 된 자식프로세스가 현재 정지되어 있다면 참을 반환한다.

[**WSTOPSIG**](https://www.joinc.co.kr/modules/moniwiki/wiki.php/manSearch?google=none&name=WSTOPSIG)**(status)**

자식을 정지하도록 야기한 신호의 숫자를 반환한다. WIFSTOPPED가 non-zero 일경우에만 사용할수 있다.

* **hint** : if 구문에 위의 매크로를 이용하여 조건을 완성할 것. **WTERMSIG**와 **WSTOPSIG**는 printf 에서 사용하는데 한번 생각해볼 것.
* **hint :** 자식 프로세스를 목록에서 제거 – **deletejob 함수** 이용

job의 상태를 STOP으로 바꿔줌– **getjobpid(jobs, pid) -> state = ST;**

코드 작성이 완료된 후에 어떤 흐름으로 돌아가는지 확인해 볼 것.

시그널의 처리 과정에 대해

이해하고 위 코드의 구현과 비교해봐야함!!