Int mai"EA0014: 유닉스프로그래밍" 실습 지침서

- 실습 번호: lab-08 이름: \_\_\_\_\_\_\_\_\_서우일\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- 실습 날짜: 2018년 05 월 28 일

- 실습 제목:  Process environment 2 학번: \_\_\_\_\_\_\_201311126\_\_\_\_\_\_\_\_\_\_\_\_\_\_

- 실습 내용:

주의: 모든 프로그램은 아래의 macro를 사용하여 시스템 호출 후 오류가 발생하면 "시스템호출이름: 오류메시지"를 표준 오류로 출력하고 errno를 exit 값으로 종료한다.

#include <stdlib.h>

#include <errno.h>

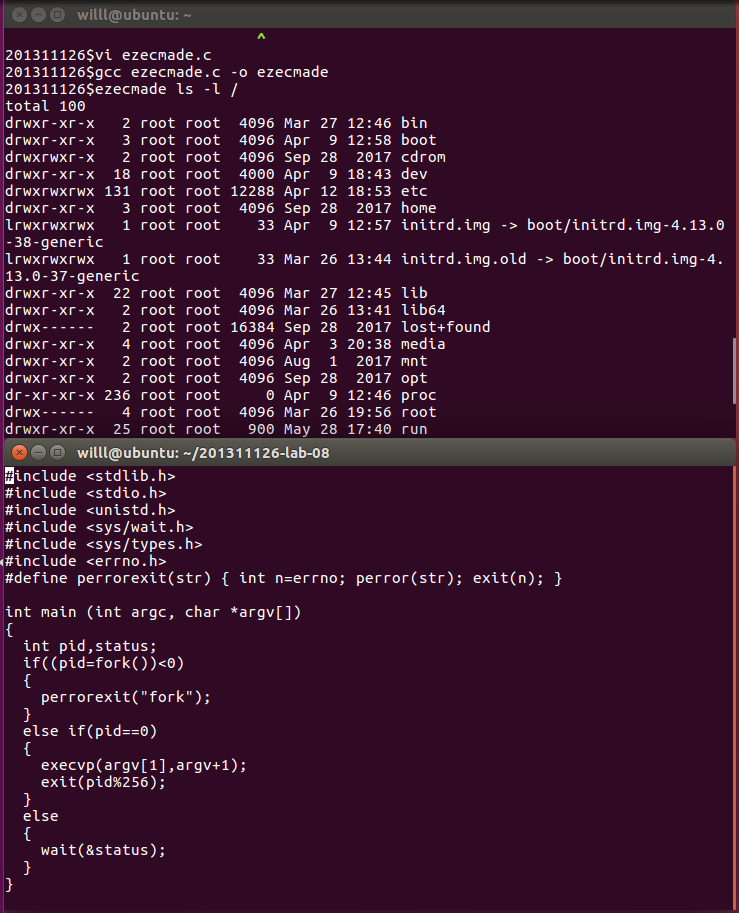
#define perrorexit(str) { int n=errno; perror(str); exit(n); }

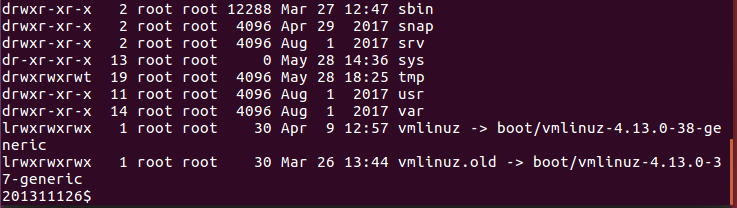
1. (실습준비) 리눅스 가상 머신을 실행하여 Login한 후 자신의 학번이 포함되도록 쉘 프람프트를 변경하라. 또한 자신의 홈 디렉토리 아래 실습 디렉토리 ~/201611234-lab-08을 만든다. 이번 실습에서 만들어지는 모든 파일들은 이 디렉토리에 두도록 한다.
2. (wait-normal.c) 다음 조건을 만족하는 프로그램 wait-normal.c를 작성하고, 실행 결과를 화면 캡처하여 삽입하시오.

* 이 프로그램은 fork() 시스템 콜을 통해 새 프로세스를 생성한다. 새로 만들어진 자식 프로세스는 exit() 시스템 콜을 통해 정상 종료하되, getpid()를 통해 자신의 프로세스 id 값을 구한 뒤, 이를 256으로 나눈 나머지 값을 exit status 로 하여 종료한다.
* 한편 부모 프로세스는 wait() 시스템 콜을 통해 자식 프로세스의 종료를 기다린 후, 매크로 WIFEXITED() 와 WEXITSTATUS()를 사용하여 자식 프로세스의 exit status 값이 제대로 전달되었는지 확인한다.



1. (exec.c) argv[1], ..., argv[argc-1]에 주어진 명령어 한 줄을 child 프로세스에서 수행시키는 프로그램을 작성하라. 예를 들어 "exec ls -l /"를 치면 "ls -l /"의 수행 결과가 화면에 출력된다. 이 프로그램을 적절한 명령어 줄을 넣어서 수행한 결과를 화면 캡쳐하여 삽입하시오. (주: 6개의 exec 관련 호출 중 어떤 것을 사용하는 것이 가장 편리한지 생각해 볼 것.)



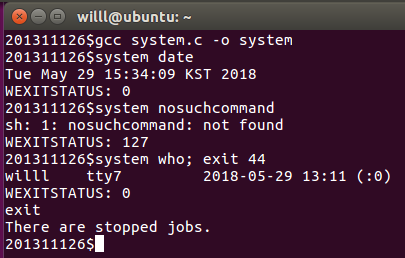


짤린 결과의 나머지입니다.

execvp를 사용하여 명령 옵션 갯수가 많은 것을 벡터로 받는 것을 이용해 쉽게 처리하기위해 이것을 사용했습니다.

1. (system.c) 수업 시간에 다룬 system() 함수를 이용, 세 개의 명령줄 “date”, “nosuchcommand”, “who; exit 44”를 각각 수행하는 프로그램을 작성하시오. 이 과정에서 매 번 system()이 리턴하는 값에 WEXITSTATUS()를 적용하여 exit status를 얻고 이를 printf()를 사용하여 화면에 출력하도록 한다. 이 프로그램의 실행 결과를 화면 캡쳐하여 삽입하고, 왜 이런 출력이 나오는지를 설명하시오.

결과입니다.



아래 페이지에 코드가 있습니다.

1. Date

date의 실행시 실행한 결과가 정상출력 정상종료 되었으므로 system함수가 리턴한 값을 WEXITSTATUS가 받아서 exit status값을 출력한 결과가 0으로 정상 종료되었음을 의미합니다.

2. Nosuchcommand

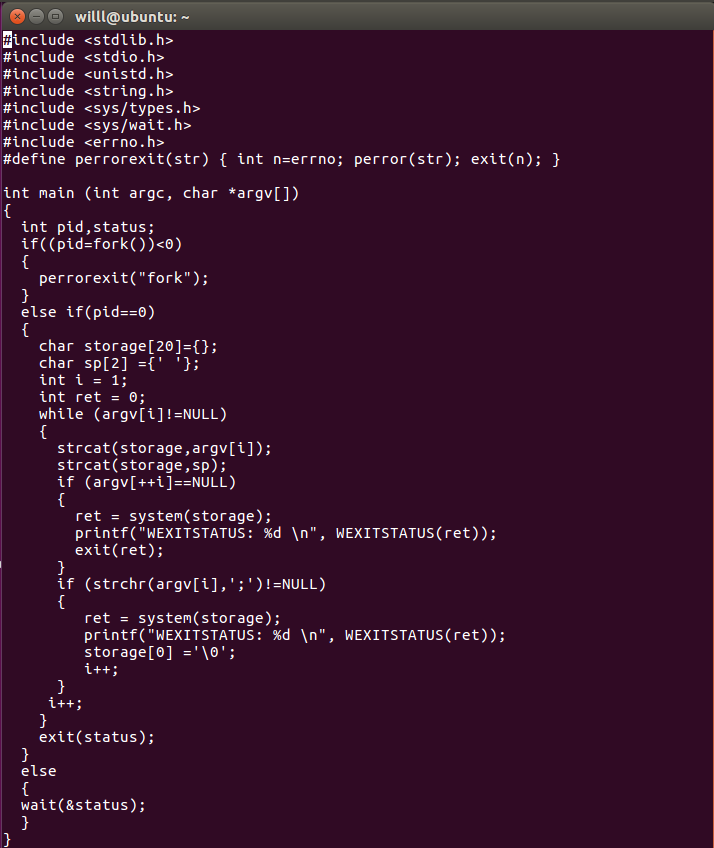
Nosuchcommand는 없는 명령어이기 때문에 오류가 발생하였고, system함수가 return한 값을 WEXITSTATUS가 받아본 결과 127이라는 번호를 반환하였습니다. 이 127의 뜻은 command not found 의 종료코드 입니다.

3. Who; exit 44

명령어 두개가 한 줄에 들어간 경우인데, 일단 who 명령은 정상실행 정상종료로 0이 나왔습니다. 그리고 뒤의 명령어인 exit 44는 제가 종료 코드에 대해 찾아본 결과,

1~2 , 126~165, 255번이 특별한 의미의 종료 코드이고, /usr/include/sysexits.h 내부의 정보에 의하면 0번, 64~78번까지가 의미가 있는 종료 코드였습니다. 즉 그냥 exit명령을 실행한 것이고, 강제종료 명령을 내린 것이라는 생각이 들었습니다.

코드입니다.



끝.