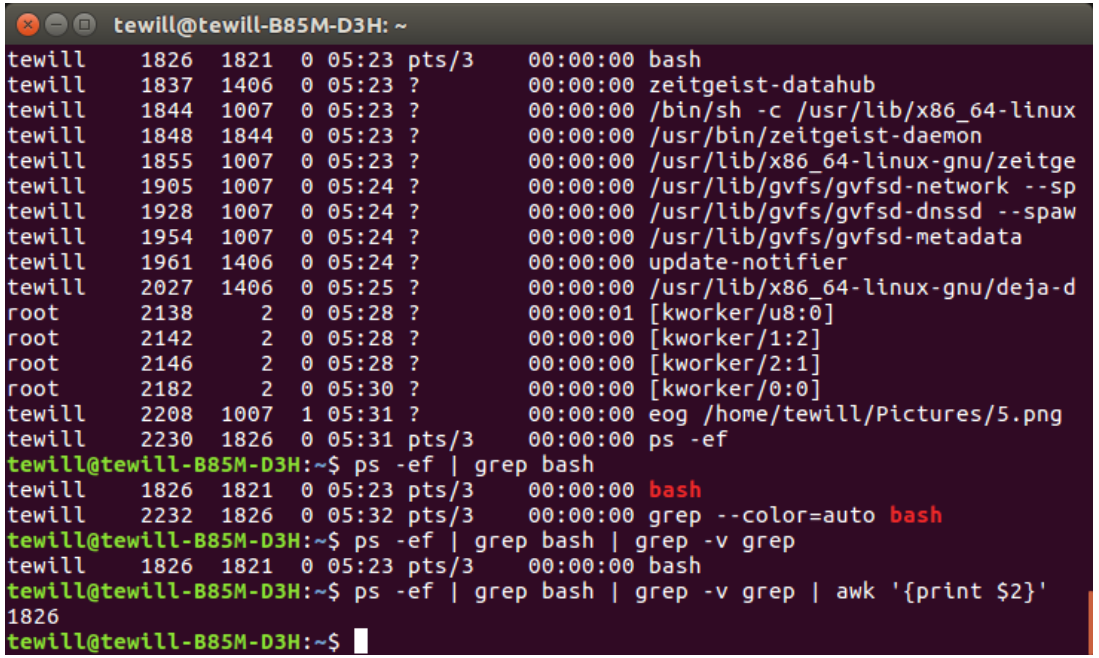
TI DSP, MCU 및 Xilinx Zynq FPGA 프로그램 전문가 과정

강사 - Innova Lee(이상훈)
gcccompil3r@gmail.com
학생 - 은태영
zero_bird@naver.com

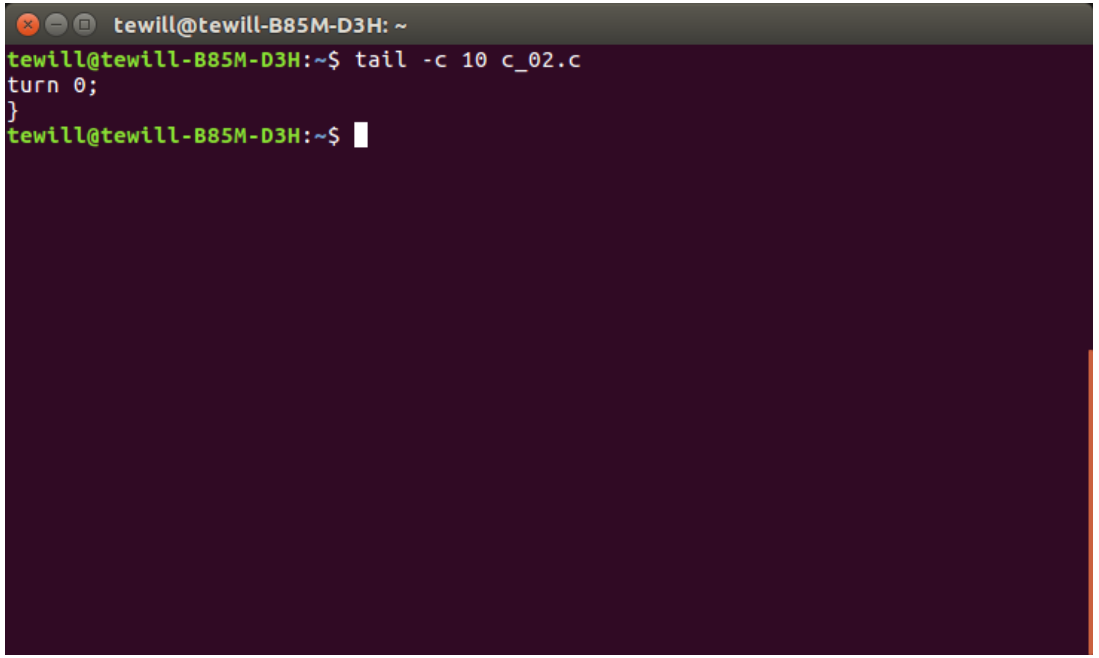
명령어



ps -ef | grep bash | grep -v grep | awk ‘{print $2}’

- ps -ef : 현제 구동중인 프로세스(main)
- | : 파이프
- grep bash(터미널) : bash(터미널)을 찾아낸다.
- grep -v grep : 자기 자신(찾는 프로세스)를 제외.
- awk ‘{print $2}’ : 2번째에 위치한 PID 출력.

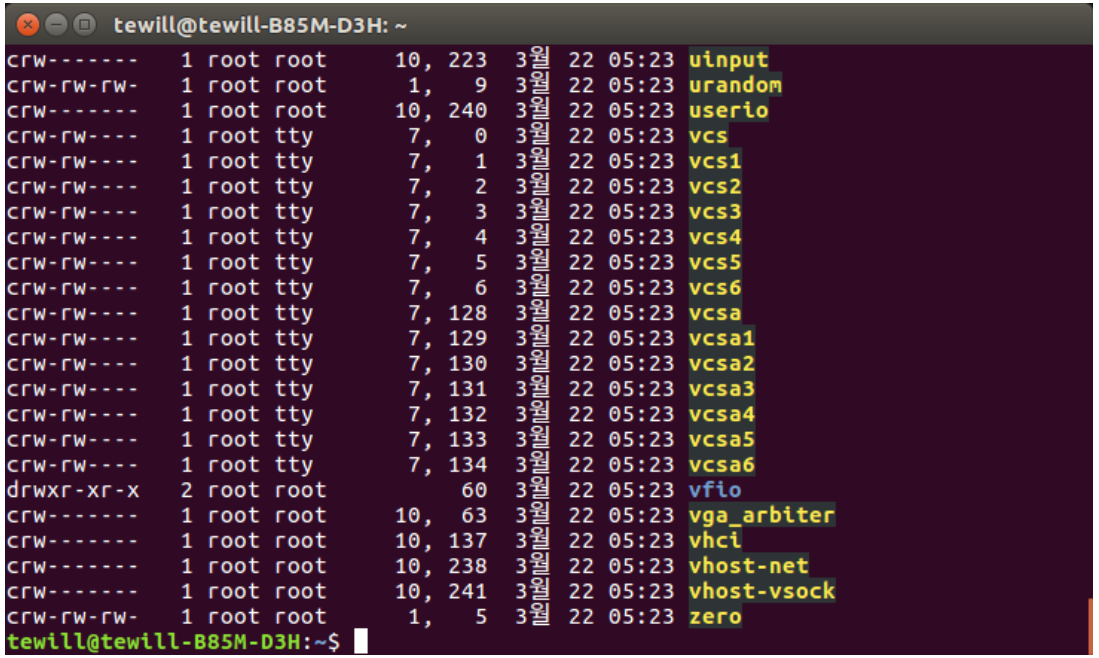
명령어



tail -c 20 oo.c

- ❖ tail : 뒤에서 마지막 부터 읽을 때.
- ❖ -c 0 : 0 번째 문자 수
- ❖ -n 0 : 0 번째 행
- ❖ /var/log/messages :

명령어



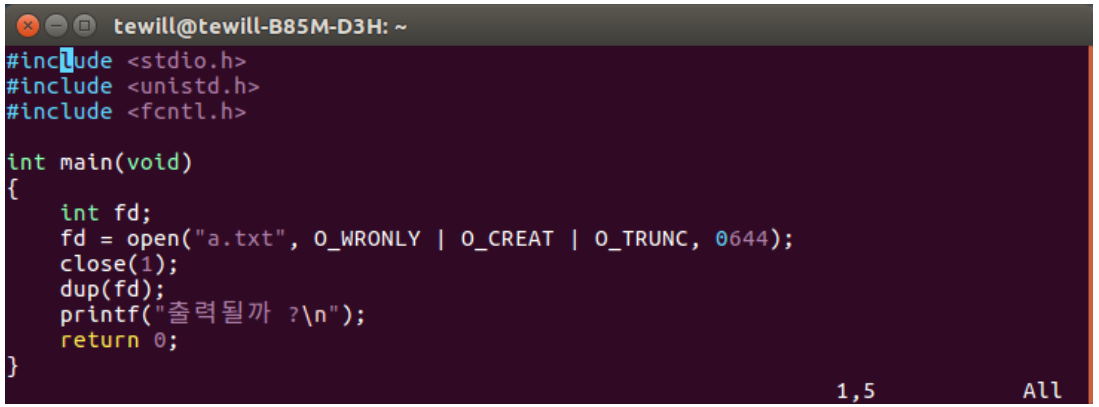
- d : 디렉토리 파일
- - : 일반 파일
- p : 파이프
- b : 블록. 특정 단위를 갖고 움직임.
- 4kb(페이지) 물리 메모리의 최소 단위.
- c : 캐릭터 디바이스. 순서. 키보드 모니터 등.
- s : 소켓. b로 통합 됨.

기타

- man 함수 : 함수에 대한 설명.
- man -s2 함수 : 시스템 콜 확인.
- atoi(문자열) : 문자열의 숫자를 정수로 리턴 한다.
- sprint(저장할 문자열, “%d”, 숫자 변수) : 숫자 등 2 번째 인자의 값을 1 번째 인자로 저장한다.
- open(파일 명, O_APPEND) : O_APPEND 는 해당 파일의 가장 끝으로 pos 를 위치시킨다.

- slab 할당자가 32byte 로 관리하기 때문에 32의 배수로 하는 것이 성능이 빠르다.

파일

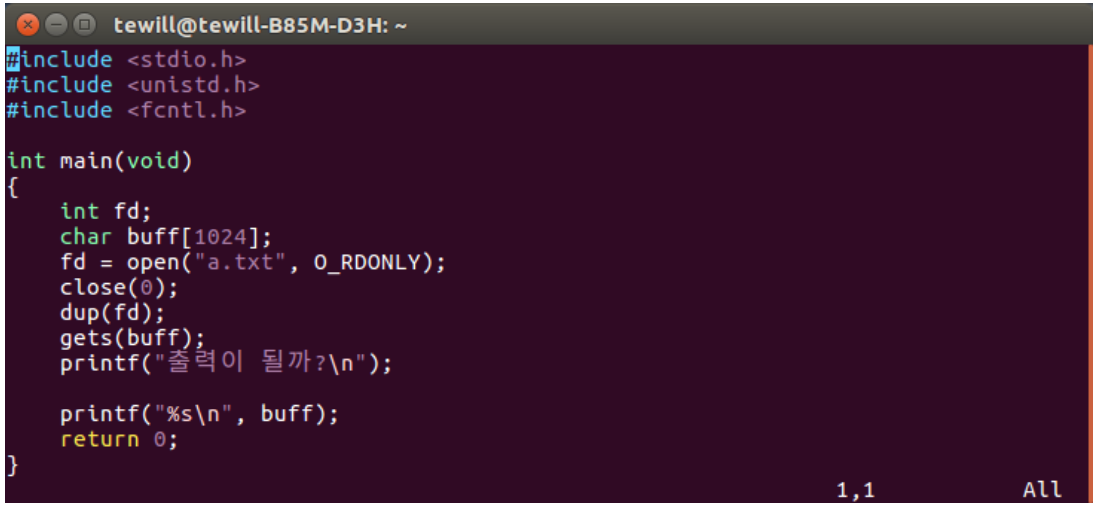


❖ dup : 가장 마지막으로 close 한 파일을 복사한다.

❖ 표준 출력을 닫고, dup 을 통해 fd 에 복사한다.

❖ printf 로 출력되야 할 문자열이 파일에 저장된다.

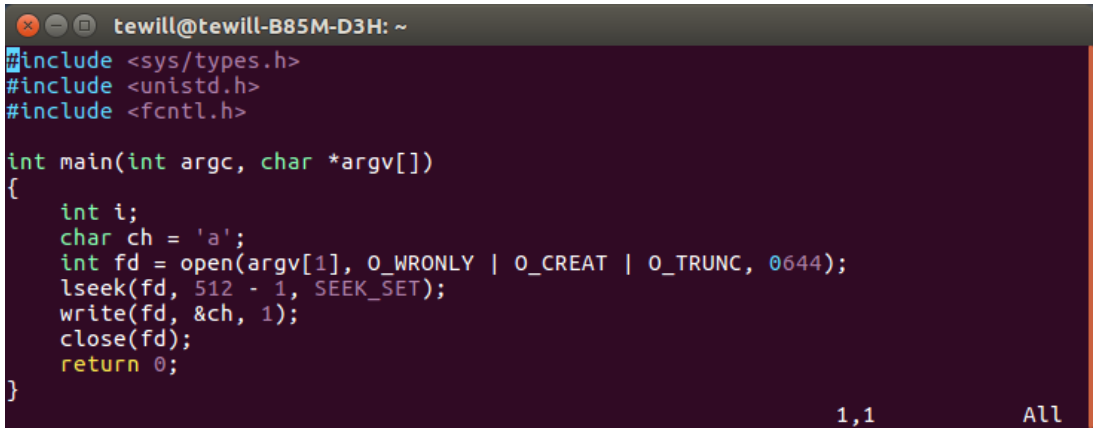
파일



- gets() : 문자열을 입력받는다.

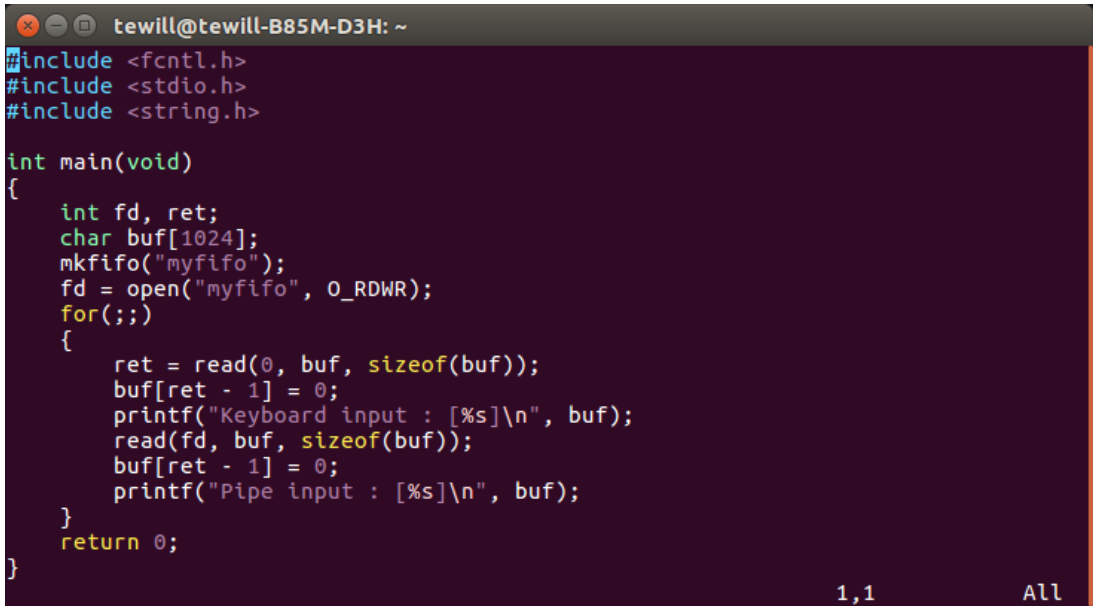
- 표준입력을 닫고 dup 을 통해 fd 에 복사한다.
- gets 를 통해 입력된 것이 파일에 저장된다.

파일



- lessk 를 통해 파일의 pos 를 조작할 수 있다.
- fd 파일의 512에 위치한 값을 a로 바꾼다.

파일



- ❖ mkfifo : 파이프 파일을 생성.
- ❖ 파이프 파일 : 데이터 값을 연결하는 통로.
- ❖ read : 블로킹. 즉, 제어권을 갖고 있다.

- ❖ 파이프 파일을 통해 통신을 하지만, read 로 인해 실행 위치에서 입력을 해야 받아올 수 있다.

- ❖ fcntl(위치, 설정, 내용) 을 통해 변경 가능하다.
- ❖ F_SETFL : 파일 권한 설정.
- ❖ O_NONBLOCK

CPU와 Multi-tasking

- CPU 는 오로지 한 순간에 한가지 연산만 수행한다.
- 1Clock 은 한 개의 연산을 하는데 걸리는 시간이다.
- CPU 주파수가 일반적으로 Ghz 단위로 동작하는데, 이는 10^9 로서 1Clock의 동작 속도는 10^{-9}가 된다.
- 아주 빠른 속도로 연산처리가 되기 때문에 Context switching 이 가능하다.
- Context switching : DRAM 에 CPU의 진행중인 연산 처리를 저장하고 다른 프로세스의 연산을 불러와 진행한다. 이때, 중요도에 따라 우선순위가 지정되며, 낮은 우선순위의 연산이 진행 안되는 것을 방지하기 위하여 할당시간을 비율로 하여, 일정 비율 이상이 될 경우 동작을 넘기도록 한다.
- 리눅스 커널의 경우, Run Queue 와 Wait Queue 가 존재하여 연산을 불러오고 실행한다.
- PIC : 하드웨어 장치의 입력을 전기 신호로 CPU 에 전달하는 역할을 한다.