Xilinx Zynq FPGA, TI DSP, MCU 기반의 프로그래밍 및 회로 설계 전문가 과정

<리눅스 시스템 프로그래밍>

2018.03.21.20일차

강사 – 이상훈
gcccompil3r@gmail.com

학생 – 안상재
sangjae2015@naver.com

- 옵션 및 명령어

- O_APPEND : 맨 뒤에 데이터를 붙임.

- man [함수]: 리눅스 커널안에 정의되어 있는 시스템 콜 함수를 출력함.

- ps -ef : 현재 cpu에서 돌아가는 프로세스의 정보를 출력함.

- ps -ef | grep bash : 'bash' 프로세스를 출력함.

- ps -ef | grep bash | grep -v grep : 'bash'를 찾는 프로세스를 제외한 나머지 프로세스들을 출력함..

- ps -ef | grep bash | grep -v grep | awk '{print $2}' : PID(process ID) 를 출력함.

- tail -c [바이트 갯수][파일명] : 파일의 맨 뒤에서부터 [바이트 갯수]만큼의 데이터를 출력함.

- tail -n [line갯수][파일명] : 파일의 맨 뒤에서부터 [line 갯수]만큼의 데이터를 출력함.

- mkfifo [파일명] : 파이프 파일을 생성함.

- cat > [파일명] : vi 편집기에서 파이프 파일을 생성함.

- ls -al : 모든 파일을 보여줌.

d~ : 디렉토리

p~ : 파이프

* c (character device) : 순서가 있음. ex) 키보드, 모니터

b (block device) : 특정 단위(메모리의 최소 단위 4KB)를 가지고 움직임. ex) HDD, DRAM

* nonblocking : 읽을 것이 있으면 읽고, 없으면 넘김 (유연성 확보)
* blocking : 어떤 일을 수행할 때까지 계속 대기상태를 유지함.

- CPU

- 프로세스는 CPU의 추상화(가상의 CPU)이다. 프로세스는 실행파일의 main문에 속하며, 하드디스크의 main의 프로그램은 기계어로 바뀌어 CPU에 전원이 들어와서 실행이 될 때 주 메모리(DRAM)로 이동되어서 실행된다.(부팅 과정)

- 컴퓨터 구조론 : CPU는 오로지 한 순간에 한 가지의 연산만 수행한다.

- CONTEXT SWITCHING

- 프로세스 : 컴퓨터 내에서 실행 중인 프로그램

- context switching은 운영체제가 multi-tasking 을 하기 위해서 꼭 필요하다.

- multi-tasking은 cpu가 한 순간에 한 가지의 연산만 수행하지만, (인간이 느끼기에) 운영체제가 cpu의 스케줄링 기법을 통해 여러 프로세스를 동시에 사용하는 것이다.

- 여러 프로세스가 돌고 있는 경우에(다중 프로세스), A 프로세스가 CPU를 점유하다가 B 프로세스에게 CPU의 점유권을 넘겨줄 때 context switching이라는 기법이 없다면 문제가 생긴다. A 프로세스가 B 프로세스에게 CPU의 점유를 넘겨주는 순간에 A 프로세스의 상태, 범용/특수 레지스터 값, pc(프로그램 카운터 값), 스케줄링 정보, 주기억장치의 정보 등을 어딘 가에 저장해 놓지 않는다면, 나중에 다시 A 프로세스가 CPU를 점유 했을 때 예전에 CPU의 점유를 넘겨줄 상태에서부터 시작할 수 없게 된다.

  따라서 프로세스가 CPU의 점유를 다른 프로세스에게 넘겨줄 때, 해당 시점의 프로세스의 정보들을 저장해 놓아야 한다. 프로세스의 정보를 저장하는 곳이 PCB (프로세스 제어 블록)이다.

  현재 CPU를 점유하고 있는 프로세스는 Run Queue에 저장이 되고, CPU를 점유하기를 기다리고 있는 프로세스는 Wait Queue에 저장이 된다.

  context switching의 단점은 파이프라인이 깨진다는 것이다.