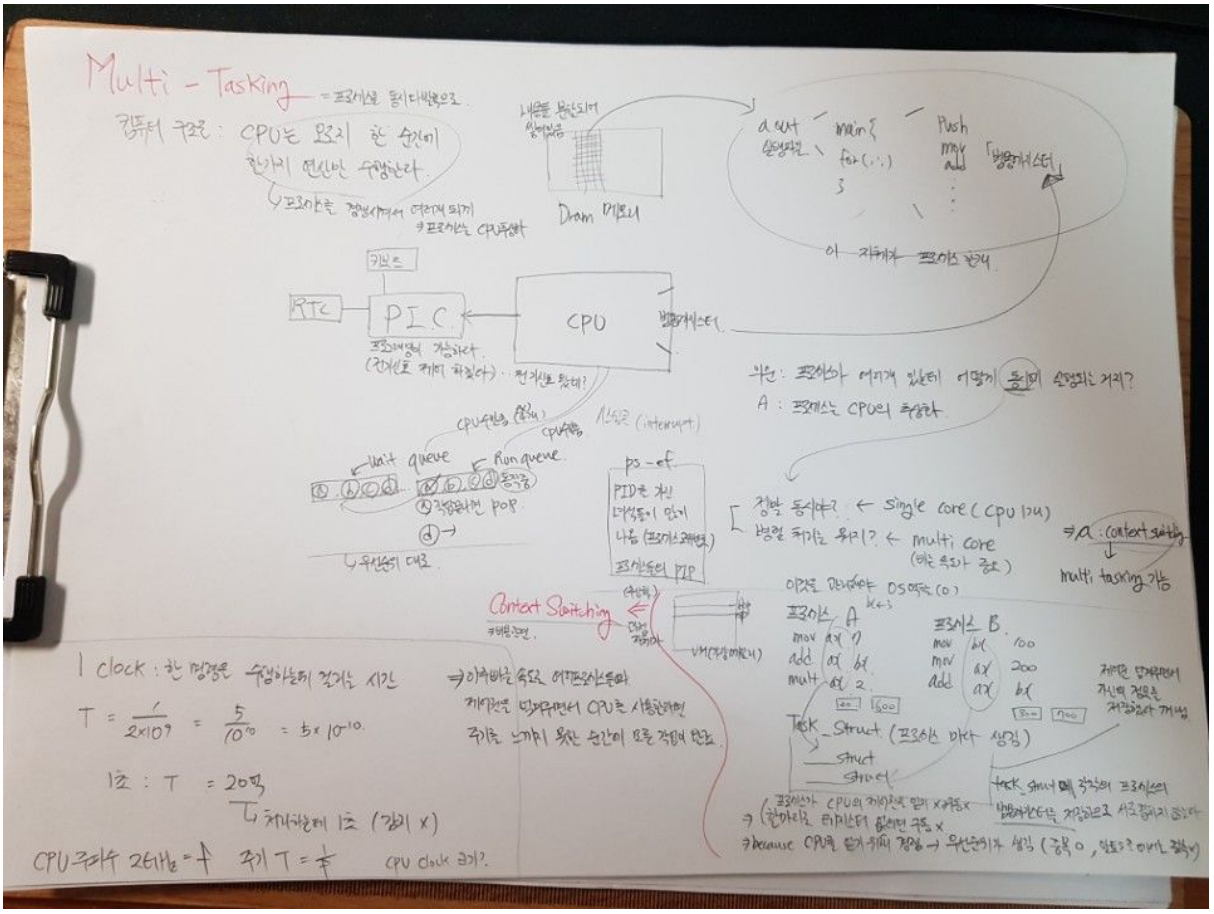
Xilinx Zynq FPGA, TI DSP,MCU기반의 프로그래밍 및 회로 설계 전문가 과정

강사 - Innov (이상훈)

gcccompil3r@gmail.com

학생 - 이유성

dbtjd1102@naver.com



principle of multi -tasking멀티테스킹의 원리

모든 cpu에 갖추고 있어야할 것 (운영체제 관점에서) = 프로세스 (프로세스는 CPU의 추상화다) (=가상의 CPU) (프로세스는 <main<기계어)

pe -ef 현재 돌아가고 있는 모든 프로세스 정보
pe -ef | grep bash bash(터미널) 찾기
pe -ef | grep bash |grep -v grep 자신을 찾는 프로세스는 제외하기
pe -ef | grep bash |grep -v grep |awk '{print $2}' 2번째 위치한 PID번호 출력(프로세스 아이디(민증))

프로세스 고유한 번호(id) =pid

tail 명령어는 파일 내용의 마지막부터 읽을때 주로 사용한다.

tail -c 20 a.txt 마지막부터 char 개수만큼 보여줌
tail -n 1 a.txt 마지막부터 line 개수만큼 보여줌

atoi(문자열) : 문자열의 숫자를 정수로 리턴
slab 할당자가 32byte로 관리하기 때문에 32의 배수로 하는것이 성능이 빠르다

principle of multi -tasking멀티테스킹의 원리

모든 cpu에 갖추고 있어야할 것 (운영체제 관점에서) = 프로세스 (프로세스는 CPU의 추상화다) (=가상의 CPU) (프로세스는 <main들이 모아진 것<기계어)

ls -al /dev ls -al 모든 파일 표시

-rw 아무것도 없는 그냥 파일
drw-directory

prw - 파이프 로컬하는데에 통신에 사용 (|)
물리 메모리의 최소단위 4kb =4096byte

crw - 캐릭터디바이스 순서가있어 (ex ..키보드 모니터 <-> 순서필요x dram (ram = random xx memory)

brw- 블록디바이스 특정사이즈로 움직임 순서필요x dram (ram = random xx memory)
,하드디스크

ls R 모든 디렉토리 순회

dup()

```
#include<unistd.h>
#include<fcntl.h>
#include<stdio.h>

int main(void)
{
        int fd;
        fd = open("a.txt", O_WRONLY | O_CREAT | O_TRUNC, 0644);
        close(1);
        dup(fd);  // duplicate 복사 fd가 1번의 역할을 해서 출력 역할을 함. fd 는
a.txt파일이니까 이 파일에 출력함.
        printf("출력될까?\n");
        return 0;
}
```

실행시 a.txt가 만들어지고 출력될까? 라는 내용이 있음.

```
#include<stdio.h>
#include<unistd.h>
#include<fcntl.h>

int main(void)

{
        int fd;
        char buff[1024];
        fd = open("a.txt",O_RDONLY);
        close(0);                   //입력x          dup는 무조건 닫고 대체가 있어야함.
        dup(fd);                    //  close한놈의 기능을  fd가 대체..              (여러개 닫혀
있었으면 최근에 닫힌거 대체)          입력자체가 파일
        gets(buff);       //입력을 받는 함수 (문자열)를 통해 입력된 것이 파일에 저장
        printf("출력될까?\n");

        printf("%s\n",buff);
        return 0;
}
```

//리바이 액션과 같은 역할 cat mytar.c >ccc:

```
#include<sys/types.h>
#include<unistd.h>
#include<fcntl.h>

int main(int argc , char *argv[])
{
        int i;
        char ch = 'a';
        int fd  = open(argv[1] , O_WRONLY | O_CREAT | O_TRUNC, 0644);

        lseek(fd,512-1,SEEK_SET);  //512 플로디스크
        write(fd,&ch,1);
        close(fd);

        return 0;
}
```

//512번째 위치에 a를 넣겠다.

//mbr master boot record 부트 코드

fifo

mkfifo : 파이프 파일을 생성
파이프 : 데이터 값을 연결하는 통로

```
#include<stdio.h>
#include<fcntl.h>
#include<string.h>
#include<unistd.h>



int main(void)
{                                   //myfifo mytito (ls쳐봐) -> 실행 -> 다른terminal에서 cat
>myfifo수행
                                  //mkfifo(" ~") 파이프(데이터 값을 연결하는 통로) 파일 생성
        int fd,ret;
        char buf[1024];
        mkfifo("myfifo");
        fd = open("myfifo",O_RDWR);

        for(;;)
        {
                ret = read(0,buf,sizeof(buf));  //read  - 블로킹 (제어권 갖고있음) - 조건이
있으면 대기
                buf[ret -1] = 0;
                printf("Keyboard input : [%s]\n",buf);
                read(fd,buf,sizeof(buf));
                buf[ret -1] = 0;
                printf("Pipe Input : [%s]\n",buf);
        }
        return 0;
}


        //블로킹은 순차적으로 진행해야 할 때에만.
        //논블로킹,SNS

        // clock 파이프라인.-->자세히 알아보기
```

//nonblocking

```
#include<stdio.h>
#include<unistd.h>
#include<fcntl.h>

int main(void)
{
        int fd,ret;
        char buf[1024];
        fd = open("myfifo",O_RDWR);          //fcntl(위치,설정,내용)을 통해 변경 가능
mkfifo 데이터값을 연결해주는 통로 ..
        fcntl(0,F_SETFL, O_NONBLOCK);  //   SETFL파일의 권한을 셋팅해 주는 것.
논블록으로  0번을 논블록으로 셋팅한다란 뜻.
        fcntl(fd,F_SETFL, O_NONBLOCK);            //마이 피포를 논블록으로 셋팅.  읽을게
없으면 걍 넘어가,,

        for(;;)
        {
                if((ret = read(0,buf,sizeof(buf))) > 0)
                {
                buf[ret-1] =0;
                printf("Keyboard input : [%s]\n",buf);
                }
                if((ret = read(fd,buf,sizeof(buf))) >0)
                {
                buf[ret-1] = 0;
                printf("Pipe input : [%s]\n",buf);
                }
        }
        close(fd);
        return 0;
}

// 이것은 mkfifo mytito          ./a.out          cat >myfifo 순서 상관 x
```