

TI DSP, MCU 및 Xilinx Zynq FPGA 프로그래밍 전문가 과정

강사 – Innova Lee(이상훈)
gcccompil3r@gmail.com

학생 – GJ (박현우)
uc820@naver.com

1. 시스템 프로그래밍 - 3 파일입출력

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

int main(void) {

    int fd;
    fd = open("a.txt", O_WRONLY | O_CREAT | O_TRUNC, 0664);
    close(1);
    dup(fd);
    printf("출력될까? \n");
    return 0;
}
```

close(1) -> 표준출력을 닫는다.

dup(fd) -> 닫은 기능을 fd가 대신한다.

printf에 "출력될까?"가 a.txt에 쓰여진다.

```
#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);
    dup(fd); // close(0)을 대체;;
    gets(buff);
    printf("출력될까?\n");
    printf("%s", buff);
    return 0;

}
```

close(0) -> 표준입력을 닫는다.

Dup(fd) -> 닫은 기능을 fd가 대신한다.

Gets()를 통해서 a.txt에 있는 내용이 buff에 입력된다.

화면에 "출력될까?"와 buff에 입력된 내용이 출력된다.

1. 시스템 프로그래밍 - 3 파일입출력

```
#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);
    write(fd, &ch, 1);
    close(fd);
    return 0;
}
```

open함수로 파일을 만든다.

lseek함수로 fd의 파일 pos를 512로 바꾸고

write로 ch값을 넣는다.

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

int main(void) {

    int fd, ret;
    char buf[1024];
    mkfifo("mvfifo");
    fd = open("mvfifo", O_RDWR);
    for(;;) {

        ret = read(0, buf, sizeof(buf));
        buf[ret - 1] = 0;
        printf("Keyboard Input : [%s]\n", buf);
        read(fd, buf, sizeof(buf));
        buf[ret - 1];
        printf("Pipe Input : [%s]\n", buf);

    }

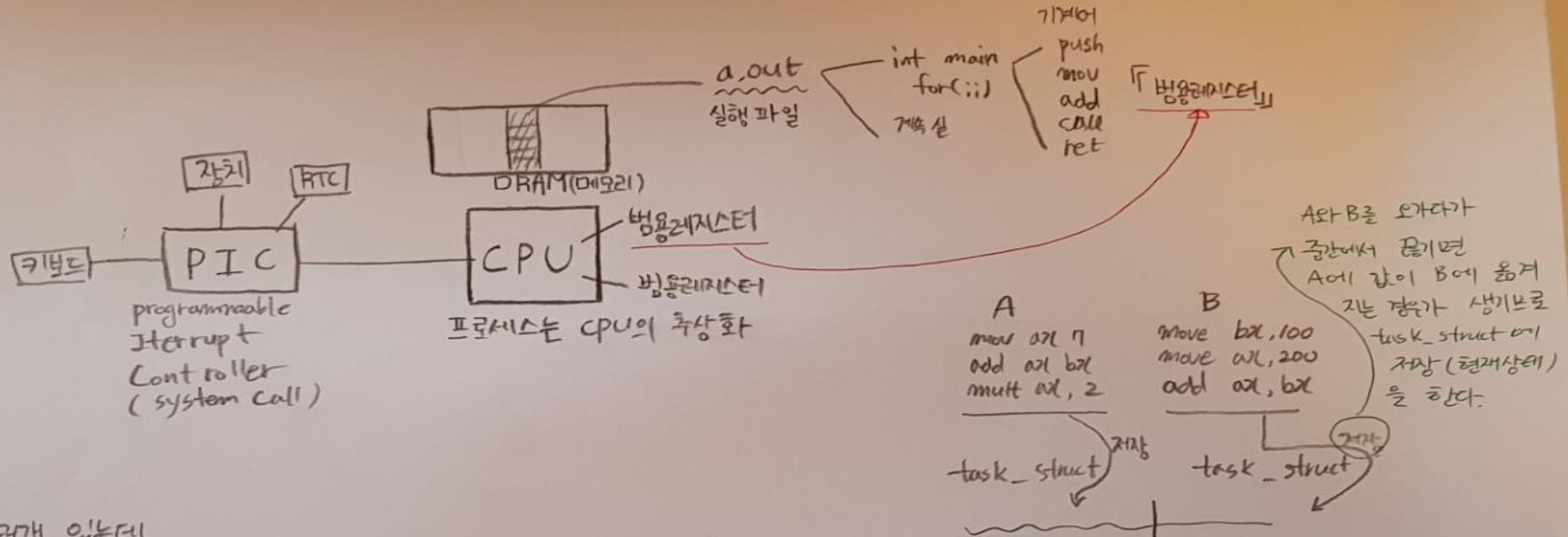
    return 0;
}
```

read는 blocking함수이기 때문에 입력만을 기다린다.

Non-blocking을 하려면 fcntl()함수를 추가한다.

Non-blocking은 입력이 언제든지 되도 상관없다.

1. 시스템 프로그래밍 - 3 파일입출력



Question

1. 프로세스가 여러개 있는데
이렇게 전부 동시에 실행되는 건가?
1) 정말 동시 실행인가? Single core

2) 병렬처리란? Multi Core

컴퓨터 구조론

→ CPU는 오로지 한 순간에
한 가지 연산만 수행한다.

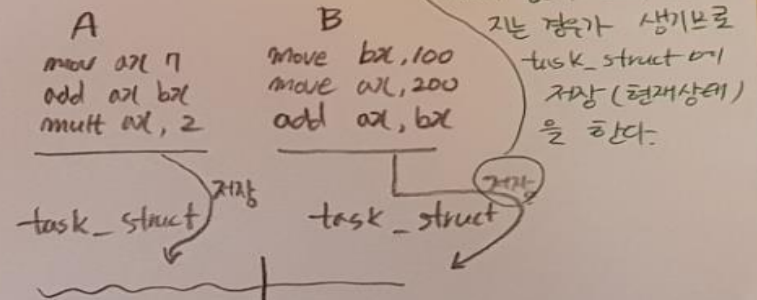
CPU 주파수: 2GHz = f (주파수)

주기: $\frac{1}{f} = T \Rightarrow 1초에 20억개$

1 clock: 몇몇이 한 개를 실행하는데
걸리는 시간

context switching

* Multi-Tasking *



A에서 B 왔다 갔다 \Rightarrow context switching
CPU 점유를 때문에 우선순위 도입

즉, 위와 같은 일련의 과정이 Multi-Tasking을
수행하는 과정이다.