**Chungnam National University**
**Department of Computer Science and Engineering**

**2010년 가을학기** **김 형신/조현우**

# 기말고사

2010년 12월 10일
시스템 프로그래밍

| 분반/학번 | 반 |
|---|---|
| 이름 | |

| 문제 | 배점 | 점수 |
|---|---|---|
| 1 | 20 | |
| 2 | 20 | |
| 3 | 10 | |
| 4 | 20 | |
| 5 | 20 | |
| 6 | 20 | |
| 총계 | 110 | |

나는 이 답안을 부정행위 없이, 내 스스로의 힘으로 작성하였으며, 다른 학생의 것을 보거나, 다른 학생에게 보여주지 않았음을 맹세합니다. (  )

문제 1. 기초지식 (20점)

1) (5점) 좀비란 무엇인가 ?

종료되었지만, 정리되지 않은(un-reaped) 프로세스

2) (5점) 동적 메모리 할당 프로그램을 작성할 때, 고려해야 하는 두 가지 중요한 성능지표를 쓰시오.

throughput and utilization

3) (5점) 가상 메모리 기능 지원을 위해 프로세서 내부에 들어 있는 MMU 모듈이 하는 일이 무엇인지 쓰시오.

가상주소를 물리주소로 번역해준다

4) (5점) worm과 virus의 차이점을 쓰시오.

worm은 스스로 실행가능하고 다른 기계에 전파하는 코드

바이러스는 자신을 다른 코드에 연결해서 동작하는 코드로 스스로 동작할 수 없다

문제 2. (10점)[보안] 다음과 같은 프로그램을 작성하였다.

```
#include <stdio.h>

char *reads(char *dest)
{
    int c = getc();
    char *p = dest;
    while (c != EOF && c != '₩n') {
        *p++ = c;
        c = getc();
    }
    *p = '₩0';
    return dest;
}

void main()
{
  char buf[64];
  reads(buf);
  puts(buf);
}
```

1) (5점) 위 프로그램을 실행하면 버퍼 오버플로우가 발생할 수 있다. 버퍼 오버플로우에 의해 손상되는 스택 내의 데이터는 무엇인가 ?

old ebp와 return address

2) (5점) 위 프로그램에서 버퍼 오버플로우가 발생하지 않도록 하려면 어떻게 프로그램을 수정하면 되는지 설명하시오.

**reads를 스트링길이를 제한할 수 있도록 수정 or 스트링 길이를 제한하는 라이브러리 사용**

문제 3. (20점)[프로세스] 다음과 같은 프로그램을 작성하였다.

```
int ccount = 0;
int N = 10;
void child_handler(int sig)
{
    int child_status;
    pid_t pid = wait(&child_status);
    ccount--;
    printf("Received signal %d from process %d\n",
          sig, pid);
    sleep(2);
}
void fork14()
{
    pid_t pid[N];
    int i, child_status;
    ccount = N;
    signal(SIGCHLD, child_handler);  /* <-------  (A)  */
    for (i = 0; i < N; i++)
       if ((pid[i] = fork()) == 0) {
            sleep(1);
          /* Child: Exit */
          exit(0);
       }
    while (ccount > 0)
       pause();/* Suspend until signal occurs */
}
```

1) (5점) 위 프로그램 (A) 줄에서 무슨 일을 수행하는지 설명하시오.

SIGCHILD 시그널 핸들러를 child_handler로 교체해서 등록해준다.

2) (5점) 이 프로그램을 실행시켰을 때 나타나는 결과를 설명하시오.

본래는 N=10으로 10번 자식 프로세스가 종료되고, 따라서 child_handler가 10번 불려서 print 문장이 10번 찍혀야 하지만, 10번보다 적은 수의 print문이 출력된다. 그래서 while 루프를 빠져나오지 못한다.

3) (5점) 이 프로그램 실행결과 의도하지 않은 결과가 발생한다면, 그 이유는 무엇인지 원인에 대해 설명하시오.

그 이유는 동일한 시그널인 SIGCHILD 를 기록하는 pending 상태를 한 개의 비트에 저장하기 때문에 연속적으로 발생하는 동일한 시그널중의 일부가 무시된다.

4) (5점) 위 2)번에서 설명한 문제를 해결할 수 있도록 위 시그널 핸들러를 수정하시오

```
void child_handler2(int sig)
{
   int child_status;
   pid_t pid;
   while ((pid = waitpid(-1, &child_status, WNOHANG)) > 0) {
      ccount--;
      printf("Received signal %d from process %d\n", sig, pid);
   }
}
```

시그널 핸들러를 위와 같이 수정한다. 단순히 sleep() 함수를 제거한다고 문제가 해결되는 것이 아니다.

문제 4. (20점) [시스템 입출력] 다음의 프로그램을 보고 문제에 답하시오.

```
int main()
{
	int fd;
	char c;

	fd = open("sample.txt", O_RDONLY, 0);

	if(fork() == 0)
	{
		read(fd, &c, 1);
		exit(0);
	}
	wait(NULL);
	read(fd, &c,1);
	printf("c = %c\n", c);
	exit(0);
}
```

1) (5 점) 이 프로그램을 실행했을 때 화면에 출력되는 내용을 쓰시오. "sample.txt"에는 abcdefg 라는 문자열이 저장되어 있다.

답) b

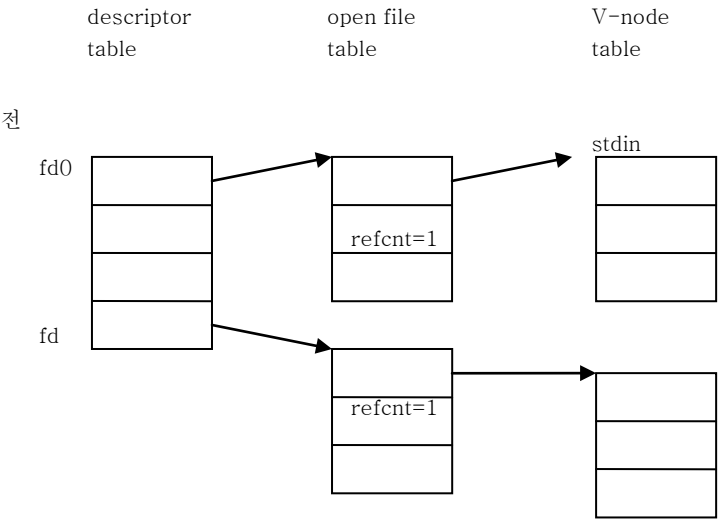
2) (5점) 리눅스 쉘 상태에서 다음과 같은 명령을 입력하면 명령의 실행 결과가 화면이 아닌 foo.txt에 저장된다. 이것을 redirection이라고 부르는데, 쉘에서 redirection을 어떤 방법을 사용해서 구현하면 되는지 설명하시오.

eslab-com$ ls > foo.txt

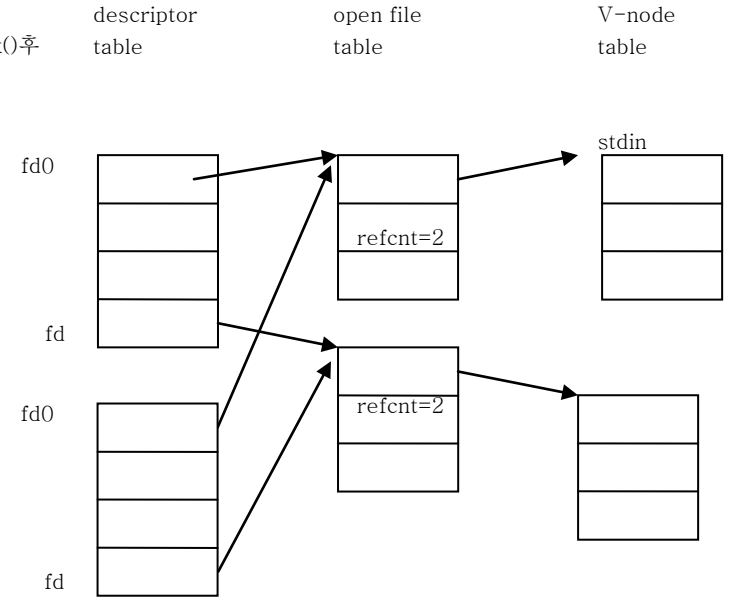
답) dup2와 같이 식별자 테이블을 조작해서 stdout파일로 가는 쓰기 작업이 특정 파일로 가도록 해준다.

3) (10 점) 위 1)번의 프로그램의 fork 실행전과 실행후의 Descriptor table, Open file table, v-node table을 각각 그리시오. 단, refcnt 값을 반드시 표시해야 하며, 포인터를 나타내는 화살표를 포함해야 한다.

(a) 실행전




(b) fork()후



문제 5. (20점) [메모리 관리]

1) (5점) 동적메모리 할당 라이브러리를 구현하기 위해 다음과 같은 alignment 조건과 블록 설계를 하는 경우에 아래표의 최소 블럭 크기를 결정하시오. ***직접리스트*** 방식으로 구현하며, payload의 크기는 0보다 커야 하고, header와 footer, pred와 succ는 각각 4바이트 워드 크기로 저장된다.

| Alignment | 할당된 블록 | Free 블록 | 최소 블록<br>크기(바이트) |
|---|---|---|---|
| Single-word | Header, footer | Header, footer,<br>pred, succ | 할당블록 최소크기<br>4+4+1 => 12<br>Free블록최소크기<br>4+4+4+4=>16<br>따라서 16바이트 |
| Double-word | Header | Header, pred,<br>succ | 할당블록 최소크기<br>4+1=>8<br>free블록 최소크기<br>4+4+4=>16<br>따라서 16바이트 |

2) (5점) 간접리스트 방식에서 Free 메모리 블럭을 찾아서 할당하려고 할 때 First Fit, Next Fit, Best Fit 의 세 가지 방식을 고려하고 있다. 이 중에서 단편화(Fragmentation)을 최소화 할 수 있는 방법은 어느 것인가 ?

답) Best Fit

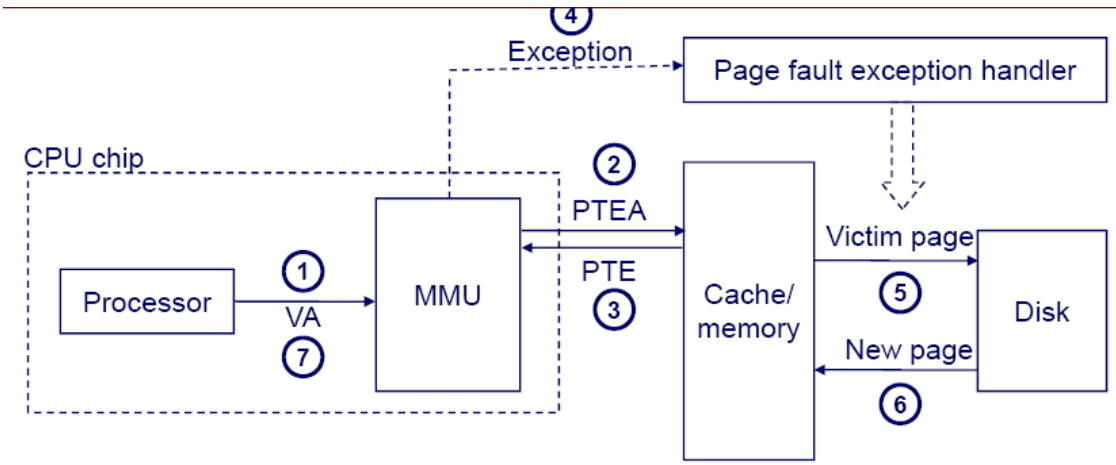
3) (5점) 양방향 포인터를 사용하는 직접 리스트 방식으로 malloc( )을 구현할 때 free블럭을 검색하는 데 소요되는 시간을 간접 리스트 방식에서와 비교해서 어느 쪽이 더 빠른지 설명하고, 그 이유를 설명하시오.

답) 직접리스트. 그 이유는 간접리스트에서는 모든 메모리 블럭을 검색해야 하지만, 직접리스트에서는 free한 블럭들만 검색하기 때문이다.

4) (5점) 직접 리스트를 사용하는 것보다 segregated list를 사용하면 얻게 되는 장점을 쓰시오.
답) 빠른 블럭 할당 및 검색 속도

문제 6. (20점) [가상메모리]

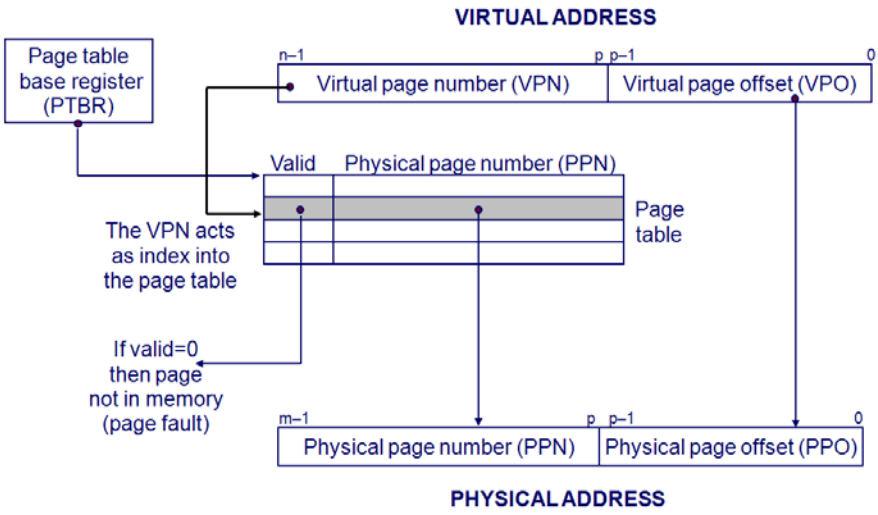
1) (5점) 아래 그림은 가상메모리 시스템에서 페이지 오류(Page Fault)가 발생하는 경우를 보여준다. 아래 그림의 5번과 6번 단계를 각각 간단히 설명하시오.



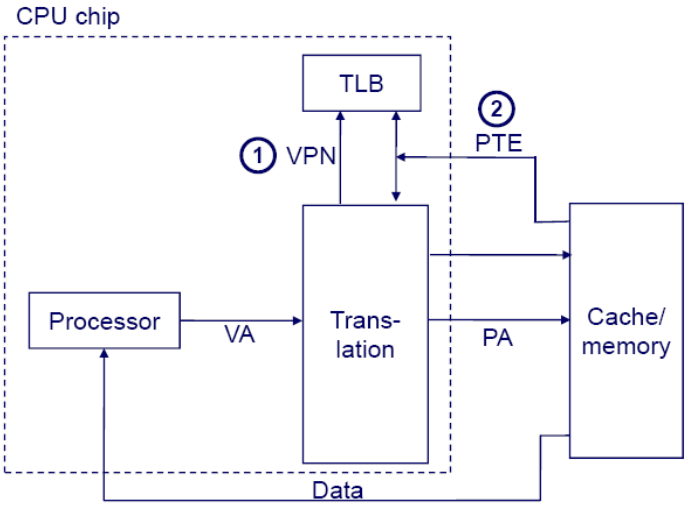

답) 5 – 메모리에서 희생자(victim) 페이지를 디스크로 복사한다

6 – 디스크에서 접근이 필요한 페이지를 메모리로 복사한다.

2) (5점) 가상 주소로부터 물리주소로 어떻게 변환되는지 그림으로 그리고 설명하시오.



3)(5점) 아래 그림은 TLB Miss가 발생했을 때 프로세서에서의 메모리 접근 과정을 보여준다. 이 그림에서 1번과 2번과정에 대해 각각 간단히 설명하시오.




**답) 1 – 가상 주소로부터 가상페이지 번호를 사용해서 TLB를 검색**
**2 – 메모리로부터 페이지 테이블 정보를 TLB로 복사**

4) (5점) 가상 메모리를 사용하면 서로 다른 프로세스의 메모리 영역을 보호해 줄 수 있다. 이를 위해서 페이지 테이블에 추가되어야 하는 항목은 무엇인가?
답) 읽기, 쓰기 허용 플래그, 접근 권한(supervisor or user)