

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

2011년 가을학기

**기말고사**

2011년 12월 12일  
시스템 프로그래밍

분반/학번	반
이름	

문제	배점	점수
1	20	
2	15	
3	10	
4	15	
5	20	
6	20	
7	10	
총계	110	

**문제 1. 기초지식 (20점)**

- 1) 버퍼 오버플로우 공격을 대비할 수 있는 방법을 한가지 제시하시오.
  
  
  
  
  
  
  
  
  
  
- 2) 운영체제의 시스템 콜(system call)이 무엇인지 설명하시오.
  
  
  
  
  
  
  
  
  
  
- 3) 자바 언어에서는 사용하지 않는 메모리를 free 시키기 위해 가비지 콜렉터를 지원한다. 교재에서 소개한 가비지 컬렉터 방법의 이름을 쓰시오.
  
  
  
  
  
  
  
  
  
  
- 4) 가상메모리를 이용해서 얻게 되는 장점 세 가지를 쓰시오.
  
  
  
  
  
  
  
  
  
  
- 5) 페이지오류(page fault)가 자주 발생하면 컴퓨터의 동작 속도가 느려진다. 그러나, 대개의 경우 page fault가 자주 발생하지 않는데, 그 이유는 프로그램의 (            ) 특성 때문이다. 괄호에 들어갈 말은 ?

문제 2. (15점)[프로세스] 다음 프로그램이 있을 때, 질문에 답하십시오.

```
int counter = 0;

int main()
{
    int i;

    for (i = 0; i < 2; i ++){
        fork();
        counter ++;
        printf("counter = %d\n", counter);
    }

    printf("counter = %d\n", counter);
    return 0;
}
```

- 1) counter = 문장은 몇 번 출력되는가?
  
  
  
  
  
  
  
  
  
  
- 2) 맨 처음으로 출력되는 counter = 값은 얼마인가?
  
  
  
  
  
  
  
  
  
  
- 3) 맨 마지막으로 출력되는 counter = 값은 얼마인가?

문제 3. (10점)[Shell] 다음은 셸에서 사용한 eval함수를 보여준다. 다음 질문에 답하시오.

```
void eval(char *cmdline)
{
    char *argv[MAXARGS]; /* argv for execve() */
    int bg;               /* should the job run in bg or fg? */
    pid_t pid;            /* process id */

    bg = parseline(cmdline, argv);
    if (!builtin_command(argv)) {
        if ((pid = fork()) == 0) {
            if (execve(argv[0], argv, environ) < 0) { /* (B) */
                printf("%s: Command not found.\n", argv[0]);
                exit(0);
            }
        }
        if (!bg) {
            int status;
            if (waitpid(pid, &status, 0) < 0) /* (A) */
                unix_error("waitfg: waitpid error");
        }
        else
            printf("%d %s", pid, cmdline);
    }
}
```

1) 위 프로그램의 (A)부분에서 수행하는 waitpid( )의 역할에 대해 설명하시오.

2) 위 프로그램 (B)에서 수행하는 execve( )의 역할에 대해 설명하시오.

**문제 4. (15점)[시그널]** 다음 각 프로그램에서 출력될 수 있는 가장 큰 값을 쓰시오. 시그널 핸들러가 동작하는 동안에는 동일한 타입의 시그널이 발생하면 처리가 블록되지만, 다른 유형에 대해서는 처리를 허용한다는 점에 유의하라.

```
/* Version A */
int i = 0;

void handler(int s) {
    if (!i) {
        kill(getpid(), SIGINT);
    }
    i++;
}

int main() {
    signal(SIGINT, handler);
    kill(getpid(), SIGINT);
    printf("%d\n", i);
    return 0;
}
```

```
/* Version B */
int i = 0;

void handler(int s) {
    if (!i) {
        kill(getpid(), SIGINT);
        kill(getpid(), SIGINT);
    }
    i++;
}

int main() {
    signal(SIGINT, handler);
    kill(getpid(), SIGINT);
    printf("%d\n", i);
    return 0;
}
```

```
/* Version C */
int i = 0;

void handler(int s) {
    if (!i) {
        kill(getpid(), SIGINT);
        kill(getpid(), SIGUSR1);
    }
    i++;
}

int main() {
    signal(SIGINT, handler);
    signal(SIGUSR1, handler);
    kill(getpid(), SIGUSR1);
    printf("%d\n", i);
    return 0;
}
```

1) Version A :

2) Version B :

3) Version C :

**문제 5. (20점) [동적메모리 할당]**

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

Alignment	할당된 블록	Free 블록	최소 블록 크기 (바이트)
Double-word	Header, footer	Header, footer	
Double-word	Header	Header, footer	

2) (5점) 간접리스트(Implicit) 방식에서 Free 메모리 블록을 찾아서 할당하려고 할 때 First Fit, Next Fit, Best Fit 의 세 가지 방식을 고려하고 있다. 이 중에서 Utilization 성능지표를 극대화 할 수 있는 방법은 어느 것인가 ?

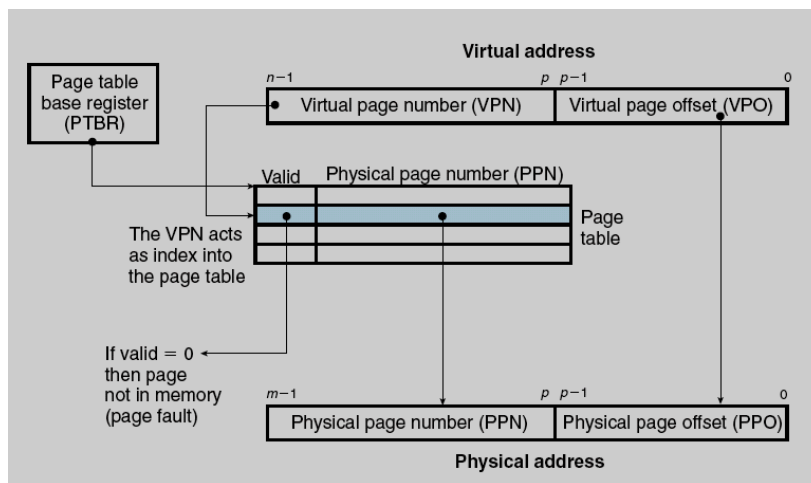
3) (5점) 양방향 포인터를 사용하는 직접리스트(Explicit list) 방식으로 malloc( )을 구현할 때 free블록을 검색하는 속도를 간접리스트(Implicit list) 방식에서와 비교할 때 어느 쪽이 더 빠르는지 설명하고, 그 이유를 설명하시오.

4) (5점) 할당된 메모리 블록을 free 시킬 때 인접블록간의 연결(coalescing)을 시도할 수 있다. 이 때, free 호출할 때마다 매번 연결을 하는 대신, 일정 시간이 흐른 뒤에 지연연결작업을 하면 Throughput과 Utilization에 어떤 영향을 줄 수 있는지 각각 설명하시오.

문제 6. (20점) [가상메모리]

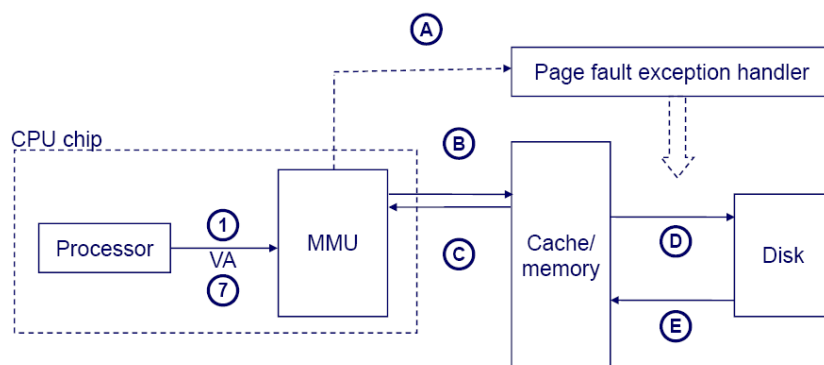
1) (4점) 어떤 가상메모리 시스템에서 가상주소의 길이가 32비트이고, 한 페이지의 크기 P가 4 KB 일 때, 이 시스템의 페이지 테이블 엔트리(PTE)의 갯수는 몇 개인가? 계산 과정도 표시하시오.

2) (8점) 아래와 같이 가상주소가 물리주소로 번역된다고 할 때, 32비트 가상주소, 24비트 물리주소를 갖는 가상메모리 시스템에서 한 페이지의 크기가 8 KB일 때 다음 각 필드의 길이가 몇 비트인지 결정하시오.



a) VPN :                      b) VPO :                      c) PPN :                      d) PPO :

3) (4점) 아래와 같이 구성된 가상메모리 시스템에서 페이지폴트(page fault)가 발생했을 때 처리되는 과정을 1번 VA(가상주소)로 부터 7번 VA 재실행까지의 중간과정 A~E가 실행되는 순서를 쓰시오.(예. ABCED)



4) (4점) 위 그림의 A~E단계를 설명하는 단어를 PTE/PTEA/Victim page/New page/Exception 중에서 각각 선택하시오.

(A)                      (B)                      (C)                      (D)                      (E)

**문제 7. (10점) [subtle error]** 다음 프로그램은 전형적인 리눅스 셸 프로그램의 구조를 보여준다. 부모 프로세스는 자신의 자식 프로세스들을 joblist 로 관리하며, 하나의 자식 프로세스에 대해 한 개의 job 엔트리를 추가한다. addjob()과 deletejob()은 이 joblist 로 부터 엔트리를 추가하거나 삭제한다. 부모 프로세스가 자식 프로세스를 생성한 후에는 이 새 프로세스를 joblist에 추가하고, 자식 프로세스가 종료되면 deletejob()을 호출해서 joblist에서 삭제한다.

```
void handler(int sig)
{
    pid_t pid;
    while ((pid = waitpid(-1, NULL, 0)) > 0) /* Reap a zombie child */
        deletejob(pid); /* Delete the child from the job list */
    if (errno != ECHILD)
        unix_error("waitpid error");
}

int main(int argc, char **argv)
{
    int pid;

    Signal(SIGCHLD, handler);
    initjobs(); /* Initialize the job list */

    while (1) {
        /* Child process */
        if ((pid = Fork()) == 0) {
            Execve("/bin/date", argv, NULL);
        }

        /* Parent process */
        addjob(pid); /* Add the child to the job list */
    }
    exit(0);
}
```

1) 얼핏 보면 잘 동작할 것 같은 위 프로그램은 미묘한 오류를 발생시킬 수 있는데, 어떤 경우에 이 오류가 발생하는지 설명하시오.

2) 이 오류를 해결할 수 있는 방법을 제시하시오