

운영체제론 중간고사'22

소프트웨어학부

2022년 4월 19일

1. [40 pts] 다음 글을 읽고 빈 칸에 알맞는 말을 채워라.

- (1) 인터럽트 벡터 테이블에는 ISR(interrupt service routine)의 _____가 저장되어 있다.
- (2) ISR이 서비스를 수행하기 전에 일반적으로 먼저 하는 일은 _____이다.
- (3) 사용자 모드에서 커널 모드로 변경될 때 런타임 스택 프레임도 변경된다. (참/거짓)이다.
- (4) 프로세스간 통신에는 공유메모리(shared memory) 방식과 _____ 방식이 있다.
- (5) 다음 코드를 실행하면 _____ 개의 프로세스가 새로 생성된다.

```
int i;
```

```
for (i = 0; i < 4; ++i)  
    fork();
```

- (6) 부모 프로세스와 자식 프로세스는 전역변수를 공유할 수 있다. (참/거짓)이다.
- (7) 스레드는 함수 내부에 선언된 지역변수를 공유할 수 있다. (참/거짓)이다.
- (8) 하드웨어 스레드는 CPU가 메모리에서 데이터를 읽거나 쓸 때 발생하는 _____ 현상을 활용하는 기술로 각 스레드는 별도의 _____를 가지고 있다.
- (9) 스케줄링 알고리즘을 평가하는 기준으로 사용하는 대기시간(waiting time)은 프로세스가 _____에 머무는 시간으로 정의한다.
- (10) 비선점(non-preemptive) 스케줄링 알고리즘은 _____의 경우를 제외하고는 CPU를 놓지 않는다.
- (11) 다중코어 환경에서 부하균형(load balancing)은 캐시 미스를 줄이기 위해 _____를 최대한 유지하는 방향으로 이루어져야 한다.
- (12) 다중코어 환경에서는 입출력 장치의 대기열을 공유하기도 하고, 코어별로 따로 유지하기도 한다. (참/거짓)이다.
- (13) 다중코어 환경에서 대기열을 공유할 경우 (경쟁상태(race condition) / 부하균형) 문제를 고려해야 한다. 선택하라.
- (14) 리눅스의 O(1) 스케줄링 알고리즘은 우선순위가 높은 작업부터 실행한다. 우선순위가 낮은 작업은 높은 작업이 모두 완료될 때까지 기다려야 한다. (참/거짓)이다.
- (15) 리눅스의 CFS(completely fair scheduling)에서 각 작업의 타임퀀텀(time quantum)은 총 합이 타겟지연(target latency)을 넘지 않도록 균등하게 할당한다. (참/거짓)이다.
- (16) CFS에서 새 작업은 가상실행시간(virtual run time)이 최저이다. 따라서 우선순위가 가장 높다. (참/거짓)이다.
- (17) CFS의 가상실행시간은 작업의 나이스(nice) 값이 0 보다 작으면 느리게 증가하고, 크면 빠르게 증가한다. (참/거짓)이다.

- (18) CFS는 새 작업이 들어와도 현재 실행 중인 작업이 타임퀀텀을 다 쓸 때까지 스케줄링하지 않는다. 따라서 CFS는 비선점 방식이다. (참/거짓)이다.
- (19) 윈도우 운영체제에서는 가변(variable) 클래스에 속한 스레드가 I/O를 기다릴 경우 우선 순위를 높여 준다. (참/거짓)이다.
- (20) 윈도우 운영체제는 디스크 I/O 보다 키보드 I/O의 우선순위가 높다. (참/거짓)이다.
2. [10 pts] 부모 프로세스가 자식 프로세스를 생성하여 명령문 `gcc -o foo foo.c`를 실행하게 하고, 자식 프로세스가 종료될 때까지 기다리는 코드를 작성하려고 한다. 주어진 `argv` 이외에 다른 변수가 필요하지 않지만 새로운 변수 하나를 추가할 수 있다. 문제에서 요구하는 것 이외에 다른 것은 고려하지 않는다. 아래 물음에 답하라.

```
#define MAXLINE 80
```

```
int main(void)
{
    char *argv[MAXLINE/2+1];

    // 여기를 완성하세요

}
```

- (a) `execv()` 함수를 사용하여 구현하라. 이 때 `gcc`의 경로는 `/usr/bin/gcc`라고 가정한다.
- (b) `execvp()` 함수를 사용하면 해는 어떻게 달라지나?
3. [10 pts] 아래 프로그램은 무작위 정수를 화면에 연속해서 출력한다. 이 프로그램을 참조해서 부모 프로세스가 무작위 정수를 생성하여 자식 프로세스에게 하나씩 넘겨주면 자식 프로세스는 그 수가 홀수인지 짝수인지를 화면에 출력하는 프로그램을 파이프(pipe)를 사용하여 작성하라. 시스템 호출시 발생할 수 있는 런타임 오류 검사는 생략해도 무방하다.

```
int main(void)
{
    srand(20220419);
    while (1)
        printf("%d ", rand());
    return 0;
}
```

힌트: 파이프 입출력은 `read()`, `write()` 시스템 호출을 사용한다.

```
ssize_t read(int fd, void *buf, size_t count);
ssize_t write(int fd, const void *buf, size_t count);
```

4. [10 pts] 다음은 스도쿠 퍼즐이 올바른지 검사하는 코드의 일부이다. 우리는 OpenMP를 사용하여 9개의 스레드가 행을 하나씩 검사하게 하려고 한다. 행 번호는 0부터 시작하며, i 번 행이 올바르면 `valid[i]`에 1을, 그렇지 않으면 0을 기록한다. 주어진 파라미터와 필요하다면 새로운 변수를 추가하여 코드에서 누락된 부분을 완성하라. 반드시 OpenMP 지시문을 사용해야 하며, 문제에서 요구하는 것 이외에 다른 것은 고려하지 않는다.

```

/*
 * 스도쿠 퍼즐이 올바르게 구성되어 있는지 스레드를 생성하여 검증한다.
 */
void check_sudoku(int sudoku[9][9], int valid[9])
{
    ...

    /*
     * 9개의 스레드를 생성하여 각 스레드가 행을 하나씩 검사한다.
     */

    // 여기를 완성하세요
    ...
}

```

5. [10 pts] 다음 스레드 생성 프로그램을 읽고 물음에 답하라.

<pre> #define N 8 int n = 0; int x; void *foo(void *arg) { int y; x = y = n++; while (x == y) /* do nothing */; printf("<%d,%d>\n", x, y); pthread_exit(NULL); } </pre>	<pre> int main(void) { int i; pthread_t tid[N]; for (i = 0; i < N; ++i) pthread_create(&tid[i], NULL, foo, NULL); for (i = 0; i < N; ++i) pthread_cancel(tid[i]); for (i = 0; i < N; ++i) pthread_join(tid[i], NULL); return 0; } </pre>
--	---

- (a) 이 프로그램을 실행하면 결과가 어떻게 되나? 실행 시나리오 하나를 제시하라.
- (b) 스레드가 지연철회(deferred cancellation) 방식을 따른다면 위 프로그램은 종료가 되지 않는다. 그 이유는 무엇인가?
- (c) 함수 `foo()`를 어떻게 고치면 위 프로그램을 종료시킬 수 있는가? 그 이유는 무엇인가?
- (d) ★ 비동기식(asynchronous)철회는 스레드를 즉시(immediately) 철회한다고 되어 있다. 그 시점이 CPU 스케줄링 상 언제라고 생각하는가? 그 이유는 무엇인가? 단, 단일 코어 환경으로 한정해서 답한다.
6. [10 pts] 프로세스의 도착시간, 주기, CPU 버스트, 마감시간이 표와 같다. 이 주기적인 프로세스를 비율단조(rate monotonic)로 스케줄링하는 것이 불가능함을 간트차트를 이용하여 보여라.

Process	Arrival	Period	CPU Burst	Deadline
P_1	0	50	25	50
P_2	0	80	35	80

HK