

운영체제론 중간고사'21

소프트웨어학부

2021년 4월 22일

1. [40 pts] 다음 글을 읽고 빈 칸에 알맞는 말을 채워라.

- (1) 인터럽트는 하드웨어 또는 소프트웨어에 의해서 발생할 수 있다. (참/거짓)이다.
- (2) 인터럽트 벡터 테이블에는 ISR(Interrupt Service Routine)의 _____가 저장되어 있다.
- (3) C 함수의 매개변수(parameter)는 _____ 공간에 할당된다.
- (4) 리눅스에서 현재 커널의 버전을 확인하는 명령어와 옵션은 _____이다.
- (5) 리눅스의 PCB(Process Control Block)에서 프로세스들은 단일 연결 리스트(singly linked list)로 연결되어 있다. (참/거짓)이다.
- (6) 한 프로세스가 다른 프로세스의 실행을 관찰하고 제어할 수 있게 하는 리눅스의 시스템 호출은 _____이다.
- (7) 다음에 있는 OpenMP 코드를 하드웨어 스레드가 8개인 쿼드코어 CPU에서 실행했다면, 그 결과는 어떻게 되나?

```
x = 0;
#pragma omp parallel
{
    x = x + 1;
}
printf("%d\n", x);
```

- (8) 다중프로세서 환경에서 로드 밸런싱은 캐시 미스를 줄이기 위해 _____를 최대한 유지하는 방향으로 이루어져야 한다.
- (9) pthread_create()으로 생성된 스레드는 지연철회(deferred cancellation)가 기본(default)이며, 모두 철회지점(cancellation point)을 가지고 있다. (참/거짓)이다.
- (10) pthread_testcancel() 함수는 값을 리턴하지 않으며, 호출한 지점으로 돌아오지 않는다. (참/거짓)이다.
- (11) 리눅스나 macOS 같은 현대의 운영체제에서 스레드의 스케줄링 경쟁 범위는 (프로세스 내/전체 시스템)이다. 선택하라.
- (12) 윈도우 운영체제에서는 가변(variable) 클래스에 속한 스레드가 I/O를 기다릴 경우 우선 순위를 높여 준다. (참/거짓)이다.
- (13) 스레드는 프로세스와 달리 데이터와 런타임 스택을 공유할 수 있다. (참/거짓)이다.
- (14) 우선순위 스케줄링 알고리즘의 가장 큰 단점은 순위가 낮은 프로세스가 지속적으로 외면 당하는 _____ 현상이다.
- (15) _____는 가장 일반적인 CPU 스케줄링 알고리즘이지만 가장 복잡한 알고리즘이기도 하다.

- (16) 하드웨어 스레드는 CPU가 메모리에서 데이터를 읽거나 쓸 때 발생하는 _____ 현상을 활용하는 기술로 각 스레드는 별도의 _____를 가지고 있다.
- (17) 스케줄링 알고리즘을 평가하는 기준으로 사용하는 대기시간(waiting time)은 프로세스가 _____에 머무는 시간으로 정의한다.
- (18) 비율단조(rate-monotonic) 스케줄링 알고리즘은 비선점(non-preemptive) 방식이다. (참/거짓)이다.
- (19) 비율단조 알고리즘으로 N 개의 프로세스를 스케줄링할 때, CPU 이용률(utilization)이 _____를 넘을 수 없다.
- (20) 리눅스의 CFS는 실행이 가능한 작업 중에서 _____ 값이 가장 작은 것을 다음에 실행할 작업으로 선택한다.
2. [10 pts] 다음은 부모 프로세스가 자식 프로세스를 생성하여 명령어 `ps -a`를 `execvp()` 함수를 사용하여 실행한 다음, 그 결과를 화면에 출력하지 않고 `out.txt` 파일에 저장하기 위한 코드의 일부이다. 부모 프로세스는 자식 프로세스가 종료될 때까지 기다려야 한다. 주어진 변수와 필요하다면 새로운 변수를 추가하여 코드에서 누락된 부분을 완성하라. 문제에서 요구하는 것 이외에 다른 것은 고려하지 않는다.

```
...
int pid;
char *args[16];
int fd;
...
    fd = open("out.txt", O_CREAT | O_RDWR, 0660);
    args[0] = "ps";

    // 여기를 완성하세요

...
```

3. [20 pts] 다음은 스도쿠 퍼즐의 각 행(row)이 올바른지 검사하는 코드의 일부로 9개의 스레드를 생성하여 각 스레드가 하나의 행을 검사하는 방식으로 검증한다. 행 번호는 0부터 시작하며, i 번 행이 올바르면 `valid[i]`에 1을, 그렇지 않으면 0을 기록한다. 주어진 변수와 필요하다면 새로운 변수를 추가하여 코드에서 누락된 부분을 완성하라. 문제에서 요구하는 것 이외에 다른 것은 고려하지 않는다.

```
int sudoku[9][9];
int valid[9];
/*
 * 다음 함수는 스도쿠 퍼즐의 각 행이 올바른지 검사한다.
 * 매개변수 arg를 통해 전달된 행 번호는 0부터 시작하며,
 * i번 행이 올바르면 valid[i]에 1을, 그렇지 않으면 0을 기록한다.
 */
void *check_row(void *arg)
{
    // 여기를 완성하세요
}
```

```

/*
 * 스도쿠 퍼즐이 올바르게 구성되어 있는지 스레드를 생성하여 검증한다.
 */
void check_sudoku(void)
{
    ...

    /*
     * 9개의 스레드를 생성하여 어떤 행을 검사하는 check_row() 함수를 실행한 후,
     * 9개의 스레드가 종료할 때까지 기다린다.
     */

    // 여기를 완성하세요
    ...
}

```

4. [10 pts] 프로세스의 도착시간, 주기, CPU 버스트, 마감시간이 아래 표와 같다. EDF (Earliest Deadline First) 스케줄링을 적용했을 때, 프로세스의 처리 과정을 간트차트를 사용하여 타임 유닛 160까지 보여라. 단, 프로세스가 선택된 이유를 설명하지 않으면 답으로 인정하지 않는다.

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

HK