

지난 시험에서 출제된 문제의 일부 (주의: 예상문제가 아님)

■ 다음 물음에 답하시오. 설명이 요구되는 질문에 대해서는 짧고 간략하게 답하시오. (각 5 점)

- Android 기반의 시스템에서는 부트디스크를 이용한 스와핑(swapping)을 지원하지 않는다. 그러나 별도의 SD카드를 이용하여 스왑 공간(swap space)을 설정하는 것은 허용한다. 가능성 있는 이유로서 타당하지 않은 것은? ① 플래시 메모리의 쓰기 횟수에 대한 제약 때문에 ② 모바일 앱(mobile apps)에서는 스와핑을 요구하지 않기 때문에 ③ 원래 Android에서는 저장공간이 부족할 때 프로세스를 스와핑하는 대신에 강제 종료시키므로 ④ 스마트폰에서 부트 디스크의 저장용량이 충분히 크지 않기 때문에
- 가상 기억(virtual memory) 공간이 2^{32} 바이트, 물리 기억(physical memory)이 2^{19} 바이트인 시스템에서 페이징 기법(페이지 크기: 1 KB)을 이용하여 메모리를 관리한다고 하자. 이때 사용자 프로세스의 가상 주소(virtual address) 12345₁₆ (hexadecimal)가 프레임(frame) number 4에 매핑되었다고 한다. 이때 ① 이 가상주소가 위치한 페이지의 번호는? (10진수로) ② 메모리로 변환된 주소는 몇 번지인가? (16진수로) 단, 페이지 번호, 프레임 번호 모두 '0'부터 시작한다.
- slab allocator의 이점을 다음에서 하나 고르시오. ① 메모리를 단순한 power-of-2 allocator를 이용하여 할당 ② 커널 코드와 데이터를 효율적으로 세그먼트로 구성되게 함 ③ 복수의 작은 세그먼트를 합병하여 큰 세그먼트로 만들어줌 ④ 메모리의 fragmentation 현상이 없음
- Linux VFS 구조에서 ()란 객체(object)는 개개의 파일을 나타낸다. 여기에 들어갈 용어를 다음에서 고르시오: inode, file, superblock, dentry
- 인터럽트를 서비스하는데 있어 성능 상의 오버헤드가 발생한다. 이러한 오버헤드가 아닌 것을 하나 고르시오. ① TLB를 flush하여야 한다. 즉 TLB의 모든 기존 내용값(content)을 삭제한다. ② 프로세스의 상태를 저장하고 복원하여야 한다. ③ 명령 파이프라인(instruction pipeline)의 조정(change)에 따른 오버헤드가 발생한다. ④ preemptive kernel에서는 스케줄링을 다시 하여야 한다.
- 경우에 따라서는 가상기억(virtual memory)에의 액세스를 직접 지원하는 virtual DMA 제어장치를 사용할 수 있다. 이때 이러한 virtual DMA가 보통의 DMA와 다른 (부가적인) 기능은 무엇일까? 간단히 설명하시오.
- 두 가지 요소를 이용하는 인증방법(two-factor authentication)에 대하여: 여기에서 “요소(factor)”는 전형적으로 세 가지 종류의 접근방법으로 주어진다. 이 세 가지 요소를 제시하고 각각에 대한 예를 하나씩 보이시오.
- 클라우드 컴퓨팅 환경에서 클라우드 서버는 가상화(virtualization) 기법을 사용하는 것이 효율적이다. 가상 기계(virtual machine)를 사용하지 않으면 어떠한 문제가 야기될까?

■ demand paging 기법을 사용하는 segmentation + paging 시스템이 있다고 하자. 여기에서 가상 주소공간은 16 비트로 주어지며, 페이지 크기는 2^{12} 이고, 각 프로세스는 2 개의 세그먼트를 갖는다고 한다. 또 세그먼트의 크기는 페이지 단위로 증가한다. 페이지 테이블과 세그먼트 테이블이 아래와 같이 주어진다고 하자. (아래의 0, 1은 이진수임)

| Segment Table | | Page Table 0 | | Page Table 1 | |
|---------------|-------------------------|--------------|---|--------------|---|
| | | Frame | V | Frame | V |
| 0 → | Pointer to Page Table 0 | 101011 | 1 | 010100 | 0 |
| | Length | 001010 | 0 | 110101 | 1 |
| 1 → | Pointer to Page Table 1 | 001011 | 1 | 110100 | 0 |
| | Length | 100110 | 1 | 011001 | 0 |
| | | 001100 | 0 | 110011 | 1 |
| | | 110110 | 1 | 001001 | 0 |
| | | 111010 | 0 | xxx | 0 |
| | | xxx | 0 | xxx | 0 |

(V: Valid)

다음의 가상주소에 대하여 물리주소로 변환하시오. 만일 segment fault 혹은 page fault가 발생하는지 여부도 밝히시오. (이러한 폴트가 일어나면 주소변환은 하지 않음) (힌트: MSB는 세그먼트 번호) (20점) (1) 0001 0100 0101 0111 (2) 1100 0100 1111 1111