Chapter10. 멀티프로세스 기반의 서버 구현 내용 확인 문제

1. 다음 중 프로세스에 대한 설명을 옳지 않은 것을 모두 고르면?
2. 프로세스는 운영체제의 관점에서 프로그램의 실행 단위가 된다.
3. 프로세스도 생성 방식에 따라서 부모와 자식의 관계를 갖는다.
4. 프로세스는 다른 프로세스를 포함할 수 있다. 즉 하나의 프로세스는 자신의 메모리 영역에 또 다른 프로세스를 포함할 수 있다.
5. 자식 프로세스는 또 다른 자식 프로세스를 생성할 수 있고, 이렇게 생성된 자식 프로세스 역시 또 다른 자식 프로세스를 생성할 수 있지만, 이들은 모두 하나의 프로세스와 만 부모 자식의 관계를 형성한다.
6. Fork 함수가 호출되면 자식 프로세스가 생성되는데, 이 자식 프로세스의 특징으로 옳지 않은 것을 모두 고르면?
7. 부모 프로세스가 소멸되면 자식 프로세스도 소멸된다.
8. 자식 프로세스는 부모 프로세스의 모든 것을 복사해서 생성되는 프로세스이다.
9. 부모 프로세스와 자식 프로세스는 전역으로 선언되는 변수를 공유한다.
10. Fork 함수 호출로 생성된 자식 프로세스는 부모 프로세스가 실행한 코드를 처음부터 fork 함수가 호출된 위치까지 실행해 온다.
11. 자식 프로세스가 생성되면 부모 프로세스의 모든 것을 복사하는데, 이때 복사의 대상으로는 소켓의 파일 디스크립터도 포함이 된다. 그렇다면 복사된 파일 디스크립터의 정수 값은 원본 파일 디스크립터의 정수 값과 동일한지 확인하기 위한 프로그램을 작성해보자.

텍스트이(가) 표시된 사진

자동 생성된 설명

1. 프로세스가 좀비가 되는 경우에 대해서 설명하고, 이를 막기 위해서 어떠한 방법을 취해야 하는지 설명해보자.
2. wait() 함수 호출
3. waitpid() 함수 호출
4. signal, sigaction 함수를 통한 시그널 핸들링
5. SIGNINT에 대한 핸들러를 등록하지 않은 상태에서 Ctrl+C 키가 입력되면, 운영체제가 지정해 놓은 디폴트 이벤트 핸들러에 의해서 프로그램이 종료되어 버린다. 그러나 Ctrl+C 키에 대한 핸들러를 직접 등록하면 프로그램은 종료되지 않고 프로그래머가 지정한 이벤트 핸들러가 대신 호출된다. 그렇다면 일반적인 프로그램에서 다음과 같이 동작하도록 이벤트 핸들러 등록을 위한 코드를 구성해보자.

“Ctrl+C 키가 눌리면, 정말로 프로그램을 종료할 것인지 묻고, 이에 대한 대답으로 ‘Y’가 입력되면 프로그램을 종료한다.”

그리고 간단히 문자열을 1초당 한번 정도 반복 출력하는 프로그램을 작성해서 위의 이벤트 핸들러 등록 코드를 적용시켜보자.

텍스트이(가) 표시된 사진

자동 생성된 설명