**운영체제**

연습문제 10주차 보고서

**학번: 20175105**

**이름: 곽영주**

**과제하시기 전에 예제 꼭 한 번씩 실행해보세요!**

1. 시그널 처리는 주로 sigaction()을 사용하지만 같은 일을 수행할 수 있는 함수로 signal()이 있습니다. 이 함수에 대해 조사하고 차이점을 기술하세요.

[설명]

sigaction()은 signal() 보다 향상된 기능을 제공하고 시그널 처리를 결정하는 함수입니다.

signal()은 처리할 행동 정보로 시그널이 발생하면 호출이 될 함수 포인터를 넘겨줍니다.

sigaction()은 struct sigaction 구조체 값을 사용하여 좀 더 다양한 지정이 가능합니다.

1. [예제 10-5]를 수정하여 모든 시그널을 봉쇄하여 “don’t disturb me”를 5초동안 출력한 후 SIGINT의 봉쇄만 제외하여 SIGINT를 받을 경우 “disturb me”를 출력하는 프로그램을 작성하세요.

|  |
| --- |
| **소스 코드** |
|  |
| **실행 결과 스크린샷** |
|  |

[설명] setAll에는 모든 시그널을 포함시키고 setSigint에는 SIGINT만 포함합니다. 다음으로 모든 시그널을 봉쇄하고 5초 후 SIGINT만 봉쇄를 해제합니다. 그리고 SIGINT만 따로 핸들러를 설정하여 출력문을 실행 후 종료합니다.

1. [예제 10-6]을 수정하여 부모 프로세스가 fork로 자식 프로세스를 생성한 다음 자식 프로세스는 1부터 5까지 더한 결과를 출력한 뒤 kill 함수를 이용하여 부모 프로세스에게 알린 후 종료되도록 하세요. 부모 프로세스는 자식 프로세스의 시그널을 받고 종료해야 하며 각 프로세스는 종료할 때 화면에 종료 문구를 출력해야 합니다. (예시: 부모 – parent bye, 자식 – child bye). Wait를 사용하는 과제가 아니며 시그널의 종류는 아무거나 사용하면 됩니다.

|  |
| --- |
| **소스 코드** |
|  |
| **실행 결과 스크린샷** |
|  |

[설명] 부모프로세스는 대기상태로 만들고 자식프로세스에서 1부타 5까지 더한 후 출력하고 대기상태인 부모프로세스로 SIGINT 시그널을 보낸 후 자식프로세스는 종료합니다. 이후 대기상태를 벗어난 부모프로세스를 종료합니다. 여기서 SIGINT핸들러를 등록하여 시그널 핸들러를 실행한 후 대기상태를 벗어나게 합니다.