

컴퓨터네트워크 프로그래밍과제 2: Selective Repeat ARQ Simulation 프로그램

다음 가이드라인에 따라 Selective Repeat ARQ 흐름제어 알고리즘을 구현하세요.

** 실제 프로그램간의 통신은 하지 않습니다.

프로그램 개발 개요:

char array (혹은 String) SEND_BUFF (MAX 1024 크기)에 보낼 문자를 화면으로 최소 200문자를 키보드로 받은 후 (단 "&" 문자는 입력이지만 전송시 발생한 오류로 간주 합니다. 재전송시 7로 변환해서 보냅니다.) RECV_BUFF (1024 크기)에 윈도우 크기만큼(n) 복사 합니다. 단 n: 윈도우크기 - 개발자가 정의). 즉 수신 함수는 n개를 복사(array 에서 array)후 Sleep(1000) 함수를 이용해 1초간 기다립니다 (전송시간으로 간주). 그 다음 수신측에서는 RECV_BUFF를 윈도우 사이즈 만큼 읽으면서 윈도우에 "&" 문자가 들어 왔다면 오류로 간주하여 해당된 번호를 NAK[] 변수에 실패(="1")기록합니다. 정상일 때는 ACK[] 변수에 성공("1")를 부여합니다. 그리고 다시 Sleep(1000)을 하여 수신자가 송신자에게 1초 동안 응답을 전송한 것으로 간주 합니다. 이제 sender 측에서는 NAK[]변수의 값 혹은 ACK[] 변수의 값을 확인하고 재전송(& 대신 7로 전송)하거나 다음번 패킷들을 보냅니다. 이 부분은 Selective Repeat ARQ 의 알고리즘을 그대로 따라하시면 됩니다. 그리고 보내고 받은 패킷 검사를 위해 패킷을 화면에 출력 바랍니다.

개발시 주의 사항: 꼭 따라하지 않아도 됨. 권장사항 임.

- 전역변수 활용 바람: 예: ACK[], NAK[], SEND_BUFF[], RECV_BUFF[]
- send() / recv() 두개의 함수로 나누어 작업
- main 기능: 사용자 데이터입력을 받음/ send() 와 recv() 함수를 반복문을 이용해 번갈아 호출
- send() 필요기능: 윈도우 크기 n개 만큼 나누어 RECV_BUFF로 copy 함/ 그리고 1초간 sleep/ NAK 및 ACK 변수 체크/현재 윈도우 위치 저장/전송된 데이터 화면 출력
- recv() 필요기능: 윈도우 사이즈 만큼 읽어 각 윈도우에 에러 유무 파악/ACK 혹은 NAK 변수에 패킷 번호 기록/1초간 sleep()/수신 처리된 데이터 화면 출력/에러 데이터 출력

*** 주의: 타이머는 구현하지 않습니다. "&" 문자를 반드시 연속 입력하여 에러 상황에 대처하는 부분을 캡처하셔서 보고서 만들기 바랍니다. 즉, 동시에 2개 이상의 loss를 고려하여야 합니다.

제출일: 6월 9일

제출방법: 추후공지

제출항목: source code/실행파일/수행 했던 결과 화면 캡처 파일

반드시 파일의 이름은 : 학번-이름-소스(혹은, 실행, 결과) 로 명명함.

Upload시 file 이름: 2008봄_과제2-학번-이름.zip