[ 게임 네트워크 프로그래밍 ]

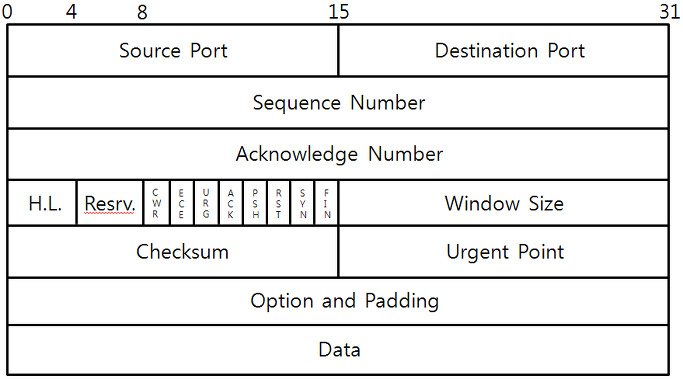
**패킷 구조 설계 및 안정성 검증**

인하대학교 미래인재개발원 문화콘텐츠프로그래밍

김예슬

**패킷 구조 설계**

- 네트워크 패킷



정보 기술에서 패킷 방식의 컴퓨터 네트워크가 전달하는 데이터의 형식화된 블록

제어 정보와 사용자 데이터로 이루어지며, 페이로드라고도 함

패킷을 지원하지 않는 컴퓨터 통신 연결은 단순히 바이트, 문자열, 비트를 독립적으로 연속하여 데이터를 전송함

데이터가 패킷으로 형식이 바뀔 때, 네트워크는 장문 메시지를 더 효과적으로 신뢰성 있게 보낼 수 있음

- 프로그램 코드

#define \_WINSOCK\_DEPRECATED\_NO\_WARNINGS

#include <WinSock2.h>

#include <iostream>

using namespace std;

#pragma comment(lib, "ws2\_32.lib")

#define PORT 7890

#define PACKET\_LENGTH 1024

#define LOCAL\_IP "127.0.0.1"

int main()

{

WSADATA wsaData;

WSAStartup(MAKEWORD(2,2), &wsaData);

SOCKET hSocket = socket(PF\_INET, SOCK\_DGRAM, IPPRPTO\_UDP);

SOCKADDR\_IN tAddr = {};

tAddr.sin\_family = AF\_INET;

tAddr.sin\_port = htons(PORT);

tAddr.sin\_addr.s\_addr = inet\_addr(LOCAL\_IP);

char Packet[PACKET\_LENGTH] = {};

strcpy\_s(Packet, "Send from Client");

sendto(hSocket, Packet, strlen(Packet), 0, (SOCKADDR\*)&tAddr, sizeof(tAddr));

SOCKADDR\_IN tDestAddr = {};

int iDestSize = sizeof(tDestAddr);

recvfrom(hSocket, Packet, PACKET\_LENGTH, 0, (SOCKADDR\*)&tDestAddr, &iDestSize);

cout << "I am Client : " << Packet << endl;

closesocket(hSocket);

WSACleanup();

return 0;

}

**모듈 안정성 검증 및 최적화**

- 소프트웨어 테스트

사용자의 요구 사항이 만족하는지 확인 및 결함을 발견하고 평가하는 활동

소프트웨어 테스트를 통해서 소프트웨어에 남아있는 결함 발견과 결함 예방 수행 가능

프로젝트 초기부터 계획하고 시작하여 제품의 품질을 더욱 더 향상시키고 소프트웨어의 유지 보수 비용을 줄일 수 있음

- 종류

1) V-Model

단위 테스트 – 통합 테스트 – 시스템 테스트 – 인수 테스트

2) 블랙박스 테스트

소프트웨어의 내부 구조를 참고하지 않고, 소프트웨어의 기능과 비 기능에 대해

테스트 수행

3) 화이트박스 테스트

블랙박스 테스트와 반대로 소프트웨어의 내부 구조를 기반으로 하여 테스트 수행