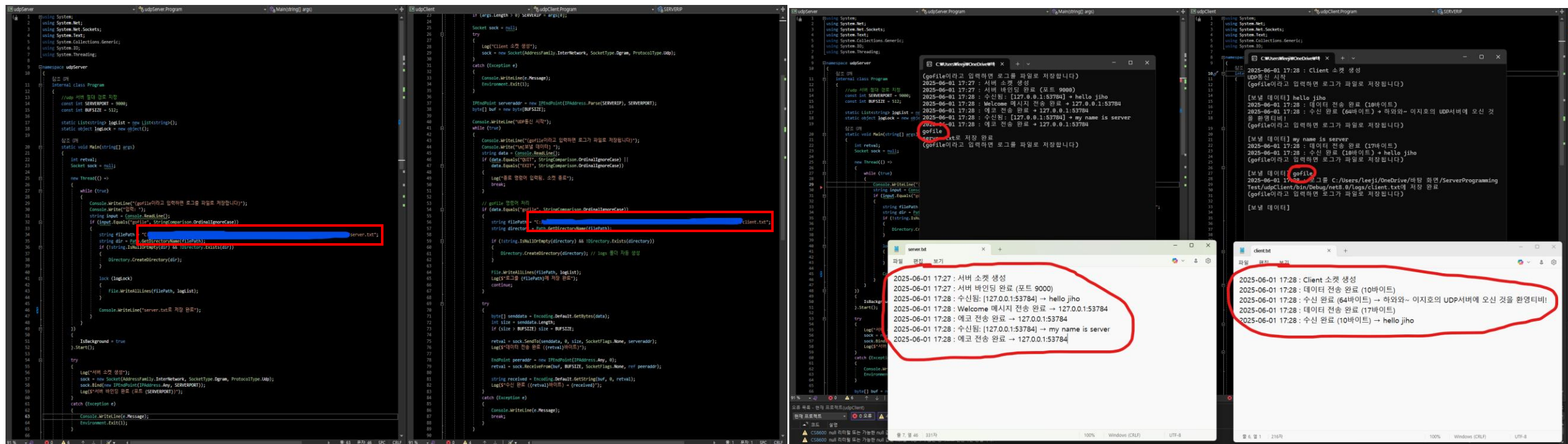


서버프로그래밍 11주차 과제

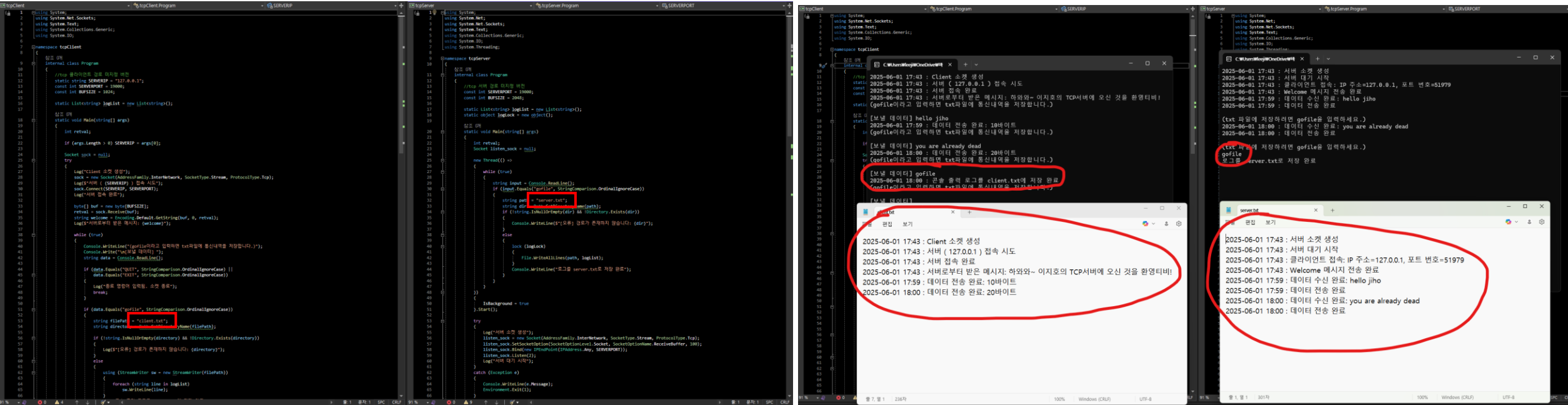
202127064 이지호

1-1. UDP 파일입출력(절대 경로 지정)



1. 서버, 클라이언트 프로그램에 각각 절대 경로를 지정
2. 통신을 하다가 gofile을 입력하면 경로에 있는 txt 파일에 통신내역을 저장

1-2. TCP 파일입출력(경로 미지정)



1. 서버, 클라이언트 프로그램에 파일명만 지정

2. 통신을 하다가 gofile을 입력하면 경로에 있는 txt 파일에 통신내역을 저장

2-1. ref 키워드 실습 예제

ref : 이미 초기화된 값을 함수에 참조로 전달하거나 함수 내부에서 값을 읽거나 쓸 때 사용됨

참조 3개

```
static void Swap(ref int i, ref int j)
```

```
{  
    int temp = i;  
    i = j;  
    j = temp;  
}
```

참조 1개

```
static void SortAscending(ref int a, ref int b, ref int c)
```

```
{  
    if (a > b) Swap(ref a, ref b);  
    if (b > c) Swap(ref b, ref c);  
    if (c > a) Swap(ref c, ref a);  
}
```

참조 0개

```
static void Main(string[] args)
```

```
{  
    int x = 5, y = 10, z = 15;  
    SortAscending(ref x, ref y, ref z);  
    Console.WriteLine(x + " " + y + " " + z);  
}
```

큰 수부터 작은 수까지 정렬하는 예제

참조 2개

```
static void ProcessFourToSum(ref int a, ref int b, ref int c, ref int d, ref int totMax, ref int totMin)
```

```
{  
    int max = a, min = a;  
  
    if (max < b) max = b;  
    if (max < c) max = c;  
    if (max < d) max = d;
```

```
    if (min > b) min = b;  
    if (min > c) min = c;  
    if (min > d) min = d;
```

```
    totMax += max;  
    totMin += min;  
}
```

참조 0개

```
static void Main(string[] args)
```

```
{  
    int a = 40, b = 5, c = 25, d = 70;  
    int totMax = 0, totMin = 0;  
  
    ProcessFourToSum(ref a, ref b, ref c, ref d, ref totMax, ref totMin);  
    ProcessFourToSum(ref a, ref b, ref c, ref d, ref totMax, ref totMin);  
  
    Console.WriteLine("최댓값 누적 결과 : " + totMax);  
    Console.WriteLine("최솟값 누적 결과 : " + totMin);  
}
```

숫자 4개를 비교해서 가장 큰 값, 가장 작은 값을 누적 연산하는 예제

2-2. out 키워드 실습 예제

out : 출력 전용 매개변수를 정의할 때 사용되며 함수 외부에서 값을 받아오고 싶을 때 사용됨

```
static bool TryCalculate(float a, float b, char input, out float res)
{
    res = 0.0f;
    if (input == '+')
    {
        res = a + b;
        return true;
    }
    else if (input == '-')
    {
        res = a - b;
        return true;
    }
    else if (input == '*')
    {
        res = a * b;
        return true;
    }
    else if (input == '/')
    {
        res = a / b;
        return true;
    }
    else
    {
        Console.WriteLine("잘못된 연산자입니다.");
        return false;
    }
}

참조 0개
static void Main(string[] args)
{
    float a = 5, b = 20;
    Console.Write("연산자를 입력하세요 (+, -, *, /): ");
    char ch = Console.ReadLine()[0];

    float res = 0.0f;
    if (TryCalculate(a, b, ch, out res))
    {
        Console.WriteLine($"결과: {res}");
    }
    else
    {
        Console.WriteLine("계산에 실패했습니다.");
    }
}
```

```
참조 1개
static void AnalyzeString(string input, out int length, out int spaceCount, out int digitCount, out int upperCount, out int lowerCount)
{
    char[] str = input.ToCharArray();
    length = str.Length;
    spaceCount = 0; digitCount = 0; upperCount = 0; lowerCount = 0;
    for(int i = 0; i < length; i++)
    {
        if (str[i] == ' ') spaceCount++;
        if (str[i] >= 48 && str[i] <= 57) digitCount++;
        if (str[i] >= 65 && str[i] <= 90) upperCount++;
        if (str[i] >= 97 && str[i] <= 122) lowerCount++;
    }
}

참조 0개
static void Main(string[] args)
{
    Console.Write("문자열을 입력하세요: ");
    string input = Console.ReadLine();

    AnalyzeString(input, out int len, out int spaces, out int digits, out int uppers, out int lowers);

    Console.WriteLine($"총 길이: {len}");
    Console.WriteLine($"공백 수: {spaces}");
    Console.WriteLine($"숫자 수: {digits}");
    Console.WriteLine($"대문자 수: {uppers}");
    Console.WriteLine($"소문자 수: {lowers}");
}
```

문자열을 입력 받고 문자열을 구성하는 요소의 수를 확인하는 예제

숫자 2개는 미리 입력하고 연산자만 입력 받아서 연산하는 예제