지능로봇실제

Assignment3

산업인공지능학과 2021254015 봉은정

1) 실행 구성

노드 구성

- node1 : 1 ~ 10 사이의 랜덤 값을 A로 publish
- node2 : 11 ~ 20 사이의 랜덤 값을 B로 publish
- node3 : A와 B를 subscribe한 후, A+B를 연산하여 C로 publish
- node4 : C subscribe 누적 평균 출력

assign 패키지 구성

- src 폴더
- CMakeLists.txt 파일
- package.xml 파일

2) 실행 결과 node1

```cpp
#include <stdio.h>
#include <stdlib.h>
#include <sstream>
#include <time.h>

#include <ros/ros.h>
#include <std_msgs/Int64.h>

int main(int argc, char** argv){

    ros::init(argc, argv, "node1");                 // 1번 노드
    ros::NodeHandle nh("~");
    ros::Publisher pubA;
    pubA = nh.advertise<std_msgs::Int64>("A", 1000); // publisher 선언 -> std_msgs::Int64 type A로 publish

    ros::Rate loop_rate(10);
    srand(time(NULL));

    while (ros::ok()) {

        std_msgs::Int64 msg;
        int randNum = rand()%10 + 1;                // 1 ~ 10 사이의 랜덤한 값

        msg.data = randNum;                         // 랜덤값을 ROS msg에 저장
        ROS_INFO("publish msg %d", msg.data);       // ROS_INFO로 msg 출력
        pubA.publish(msg);

        ros::spinOnce();
        loop_rate.sleep();
    }
    return 0;
}
```

```
ej@ej-G5-5590:~/WorkSpace/ROS_지능 로봇실제/assignment3$ rosrun assign assign
[ INFO] [1637030817.797515600]: publish msg 9
[ INFO] [1637030817.897697275]: publish msg 4
[ INFO] [1637030817.997834558]: publish msg 3
[ INFO] [1637030818.097757632]: publish msg 2
[ INFO] [1637030818.197952271]: publish msg 2
[ INFO] [1637030818.297773903]: publish msg 4
[ INFO] [1637030818.397832201]: publish msg 1
[ INFO] [1637030818.497681057]: publish msg 7
[ INFO] [1637030818.597870161]: publish msg 3
[ INFO] [1637030818.697876927]: publish msg 2
[ INFO] [1637030818.797728475]: publish msg 8
[ INFO] [1637030818.897674265]: publish msg 3
[ INFO] [1637030818.997870550]: publish msg 2
[ INFO] [1637030819.097840943]: publish msg 5
[ INFO] [1637030819.197871206]: publish msg 2
[ INFO] [1637030819.297664116]: publish msg 3
[ INFO] [1637030819.397766338]: publish msg 4
[ INFO] [1637030819.497870140]: publish msg 4
[ INFO] [1637030819.597866233]: publish msg 4
[ INFO] [1637030819.697680129]: publish msg 8
[ INFO] [1637030819.797617931]: publish msg 3
[ INFO] [1637030819.897676498]: publish msg 2
[ INFO] [1637030819.997673789]: publish msg 7
[ INFO] [1637030820.097668588]: publish msg 9
[ INFO] [1637030820.197691309]: publish msg 10
[ INFO] [1637030820.297756545]: publish msg 6
[ INFO] [1637030820.397682455]: publish msg 2
[ INFO] [1637030820.497679102]: publish msg 5
[ INFO] [1637030820.597690536]: publish msg 8
[ INFO] [1637030820.697685439]: publish msg 2
[ INFO] [1637030820.797684533]: publish msg 9
[ INFO] [1637030820.897629062]: publish msg 8
[ INFO] [1637030820.997869966]: publish msg 7
[ INFO] [1637030821.097873203]: publish msg 3
[ INFO] [1637030821.197877987]: publish msg 1
[ INFO] [1637030821.297845448]: publish msg 9
[ INFO] [1637030821.397758039]: publish msg 8
[ INFO] [1637030821.497631215]: publish msg 4
```

2) 실행 결과 node2

```cpp
#include <stdio.h>
#include <stdlib.h>
#include <sstream>
#include <time.h>

#include <ros/ros.h>
#include <std_msgs/Int64.h>

int main(int argc, char** argv){

    ros::init(argc, argv, "node2");                  // 2번 노드
    ros::NodeHandle nh("~");
    ros::Publisher pubB;
    pubB = nh.advertise<std_msgs::Int64>("B", 1000); // publisher 선언 -> std_msgs::Int64 type B로 publish

    ros::Rate loop_rate(10); // 10hz 주기
    srand(time(NULL));

    while (ros::ok()) {

        std_msgs::Int64 msg;
        int randNum = rand()%10 + 11;                // 11 ~ 20 사이의 랜덤한 값

        msg.data = randNum;                          // 랜덤값을 ROS msg에 저장
        ROS_INFO("publish msg %d", msg.data);        // ROS_INFO로 msg 출력
        pubB.publish(msg);

        ros::spinOnce();
        loop_rate.sleep();
    }
    return 0;
}
```

```
ej@ej-G5-5590:~/WorkSpace/ROS_지능 로봇실제/assignment3$ rosrun assign node2
[ INFO] [1637030824.995483853]: publish msg 12
[ INFO] [1637030825.095640149]: publish msg 20
[ INFO] [1637030825.195653410]: publish msg 18
[ INFO] [1637030825.295622428]: publish msg 19
[ INFO] [1637030825.395727587]: publish msg 20
[ INFO] [1637030825.495604624]: publish msg 11
[ INFO] [1637030825.595750593]: publish msg 17
[ INFO] [1637030825.695644654]: publish msg 12
[ INFO] [1637030825.795654834]: publish msg 17
[ INFO] [1637030825.895639231]: publish msg 11
[ INFO] [1637030825.995655117]: publish msg 18
[ INFO] [1637030826.095644382]: publish msg 16
[ INFO] [1637030826.195646795]: publish msg 13
[ INFO] [1637030826.295644091]: publish msg 17
[ INFO] [1637030826.395655312]: publish msg 16
[ INFO] [1637030826.495726311]: publish msg 14
[ INFO] [1637030826.595649870]: publish msg 13
[ INFO] [1637030826.695742079]: publish msg 14
[ INFO] [1637030826.795734473]: publish msg 14
[ INFO] [1637030826.895736779]: publish msg 15
[ INFO] [1637030826.995722103]: publish msg 15
[ INFO] [1637030827.095656990]: publish msg 18
[ INFO] [1637030827.195654011]: publish msg 16
[ INFO] [1637030827.295632521]: publish msg 11
[ INFO] [1637030827.395589977]: publish msg 13
[ INFO] [1637030827.495752635]: publish msg 19
[ INFO] [1637030827.595748735]: publish msg 12
[ INFO] [1637030827.695747753]: publish msg 17
[ INFO] [1637030827.795845764]: publish msg 12
[ INFO] [1637030827.895839477]: publish msg 19
[ INFO] [1637030827.995837610]: publish msg 12
[ INFO] [1637030828.095845915]: publish msg 15
[ INFO] [1637030828.195839977]: publish msg 20
[ INFO] [1637030828.295820870]: publish msg 11
[ INFO] [1637030828.395839901]: publish msg 14
[ INFO] [1637030828.495823206]: publish msg 19
[ INFO] [1637030828.595746636]: publish msg 14
```

2) 실행 결과 node3

```cpp
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#include <ros/ros.h>
#include <std_msgs/Int64.h>

void subscribeA(const std_msgs::Int64 msg);
void subscribeB(const std_msgs::Int64 msg);
void publishC(const ros::TimerEvent&);

int64_t A, B, C;                                    // 전역변수 A, B, C
ros::Publisher pub_c;

int main(int argc, char** argv){
    ros::init(argc, argv, "node3");                 // 3번 노드
    ros::NodeHandle nh("~");
    ros::Subscriber sub_A, sub_B;

    sub_A = nh.subscribe("/node1/A", 1000, subscribeA);   // "/node1/A" topic subscribe
    sub_B = nh.subscribe("/node2/B", 1000, subscribeB);   // "/node2/B" topic subscribe
    pub_c = nh.advertise<std_msgs::Int64>("C", 1000);     // publisher 지정 -> std_msgs::Int64 type C로 publish

    ros::Timer timer = nh.createTimer(ros::Duration(0.1), publishC); // timer 설정하여, 0.1주기로 publishC 함수 호출
    ros::spin();

    return 0;
}

void subscribeA(const std_msgs::Int64 msg) {
    A = msg.data;                                   // 전역변수 A에 msg topic 값 저장
}
void subscribeB(const std_msgs::Int64 msg) {
    B = msg.data;                                   // 전역변수 B에 msg topic 값 저장
}

void publishC(const ros::TimerEvent&) {
    C = A + B;                                      // A + B 연산
    ROS_INFO("A, B, C = %d, %d, %d", A, B, C);      // ROS_INFO로 A, B, C출력
    std_msgs::Int64 msg;
    msg.data = C;                                   // C 값을 ROS msg에 저장
    pub_c.publish(msg);
}
```

```
^Cej@ej-G5-5590:~/WorkSpace/ROS_지능 로봇실제/assignment3$ rosrun assign node3
[ INFO] [1637030832.294489386]: A, B, C = 0, 0, 0
[ INFO] [1637030832.394397726]: A, B, C = 0, 0, 0
[ INFO] [1637030832.494412255]: A, B, C = 0, 0, 0
[ INFO] [1637030832.594406987]: A, B, C = 2, 15, 17
[ INFO] [1637030832.694405579]: A, B, C = 2, 15, 17
[ INFO] [1637030832.794405342]: A, B, C = 4, 16, 20
[ INFO] [1637030832.894307967]: A, B, C = 1, 14, 15
[ INFO] [1637030832.994297341]: A, B, C = 4, 13, 17
[ INFO] [1637030833.094297618]: A, B, C = 9, 11, 20
[ INFO] [1637030833.194360916]: A, B, C = 6, 20, 26
[ INFO] [1637030833.294308582]: A, B, C = 5, 17, 22
[ INFO] [1637030833.394508829]: A, B, C = 5, 17, 22
[ INFO] [1637030833.494294260]: A, B, C = 1, 11, 12
[ INFO] [1637030833.594516973]: A, B, C = 4, 11, 15
[ INFO] [1637030833.694496087]: A, B, C = 7, 13, 20
[ INFO] [1637030833.794511026]: A, B, C = 1, 12, 13
[ INFO] [1637030833.894494760]: A, B, C = 2, 19, 21
[ INFO] [1637030833.994503531]: A, B, C = 4, 17, 21
[ INFO] [1637030834.094308698]: A, B, C = 6, 17, 23
[ INFO] [1637030834.194512416]: A, B, C = 6, 14, 20
[ INFO] [1637030834.294362296]: A, B, C = 7, 14, 21
[ INFO] [1637030834.394394997]: A, B, C = 7, 15, 22
[ INFO] [1637030834.494407807]: A, B, C = 6, 11, 17
[ INFO] [1637030834.594502067]: A, B, C = 4, 11, 15
[ INFO] [1637030834.694505767]: A, B, C = 10, 19, 29
[ INFO] [1637030834.794512361]: A, B, C = 6, 19, 25
[ INFO] [1637030834.894509123]: A, B, C = 3, 16, 19
[ INFO] [1637030834.994510691]: A, B, C = 4, 15, 19
[ INFO] [1637030835.094523099]: A, B, C = 6, 16, 22
[ INFO] [1637030835.194492269]: A, B, C = 3, 20, 23
[ INFO] [1637030835.294518523]: A, B, C = 8, 15, 23
[ INFO] [1637030835.394500923]: A, B, C = 7, 15, 22
[ INFO] [1637030835.494504353]: A, B, C = 2, 20, 22
[ INFO] [1637030835.594398526]: A, B, C = 2, 11, 13
[ INFO] [1637030835.694471679]: A, B, C = 8, 20, 28
[ INFO] [1637030835.794445335]: A, B, C = 5, 17, 22
[ INFO] [1637030835.894398406]: A, B, C = 7, 19, 26
[ INFO] [1637030835.994508301]: A, B, C = 1, 13, 14
```

2) 실행 결과 node4

```cpp
#include <ros/ros.h>
#include <std_msgs/Int64.h>

int sum, num;                                       // 전역변수 sum, num 선언
std::vector<int> array;
void subscribeC(const std_msgs::Int64 msg);

int main(int argc, char** argv){

    ros::init(argc, argv, "node4");                 // 4번 노드
    ros::NodeHandle nh("~");
    ros::Subscriber sub_C;
    sub_C= nh.subscribe("/node3/C", 1000, subscribeC); // "/node3/C" topic subscribe

    sum, num = 0;                                   // 전역변수 sum, num 초기화

    ros::spin();

    return 0;

}

void subscribeC(const std_msgs::Int64 msg) {

    int C = msg.data;
    sum += C;
    num += 1;
    double cumulativeAvg = double(sum) / double(num);                    // C의 누적평균 계산

    ROS_INFO("C : %d, sum : %d, Cumulative Average : %.2lf", C, sum, cumulativeAvg); // C의 누적평균 출력

}
```

```
^Cej@ej-G5-5590:~/WorkSpace/ROS_지능 로봇실제/assignment3$ rosrun assign node4
[ INFO] [1637030888.712507127]: C : 24, sum : 24, Cumulative Average : 24.00
[ INFO] [1637030888.812383557]: C : 26, sum : 50, Cumulative Average : 25.00
[ INFO] [1637030888.912061602]: C : 24, sum : 74, Cumulative Average : 24.67
[ INFO] [1637030889.011995047]: C : 17, sum : 91, Cumulative Average : 22.75
[ INFO] [1637030889.111994306]: C : 16, sum : 107, Cumulative Average : 21.40
[ INFO] [1637030889.212393387]: C : 18, sum : 125, Cumulative Average : 20.83
[ INFO] [1637030889.312413606]: C : 16, sum : 141, Cumulative Average : 20.14
[ INFO] [1637030889.412402178]: C : 30, sum : 171, Cumulative Average : 21.38
[ INFO] [1637030889.512392394]: C : 15, sum : 186, Cumulative Average : 20.67
[ INFO] [1637030889.612392135]: C : 25, sum : 211, Cumulative Average : 21.10
[ INFO] [1637030889.712394269]: C : 20, sum : 231, Cumulative Average : 21.00
[ INFO] [1637030889.812124143]: C : 23, sum : 254, Cumulative Average : 21.17
[ INFO] [1637030889.912350422]: C : 18, sum : 272, Cumulative Average : 20.92
[ INFO] [1637030890.012400935]: C : 22, sum : 294, Cumulative Average : 21.00
[ INFO] [1637030890.112409111]: C : 28, sum : 322, Cumulative Average : 21.47
[ INFO] [1637030890.212422497]: C : 19, sum : 341, Cumulative Average : 21.31
[ INFO] [1637030890.312431278]: C : 21, sum : 362, Cumulative Average : 21.29
[ INFO] [1637030890.412439226]: C : 19, sum : 381, Cumulative Average : 21.17
[ INFO] [1637030890.512394723]: C : 16, sum : 397, Cumulative Average : 20.89
[ INFO] [1637030890.612423336]: C : 22, sum : 419, Cumulative Average : 20.95
[ INFO] [1637030890.712384171]: C : 22, sum : 441, Cumulative Average : 21.00
[ INFO] [1637030890.812401770]: C : 29, sum : 470, Cumulative Average : 21.36
[ INFO] [1637030890.912409004]: C : 18, sum : 488, Cumulative Average : 21.22
[ INFO] [1637030891.012391438]: C : 15, sum : 503, Cumulative Average : 20.96
[ INFO] [1637030891.112397871]: C : 28, sum : 531, Cumulative Average : 21.24
[ INFO] [1637030891.212394080]: C : 20, sum : 551, Cumulative Average : 21.19
[ INFO] [1637030891.312387795]: C : 22, sum : 573, Cumulative Average : 21.22
[ INFO] [1637030891.412384886]: C : 22, sum : 595, Cumulative Average : 21.25
[ INFO] [1637030891.512346566]: C : 26, sum : 621, Cumulative Average : 21.41
[ INFO] [1637030891.612298322]: C : 17, sum : 638, Cumulative Average : 21.27
[ INFO] [1637030891.712331508]: C : 24, sum : 662, Cumulative Average : 21.35
[ INFO] [1637030891.812403063]: C : 19, sum : 681, Cumulative Average : 21.28
[ INFO] [1637030891.912343943]: C : 23, sum : 704, Cumulative Average : 21.33
[ INFO] [1637030892.012439298]: C : 26, sum : 730, Cumulative Average : 21.47
[ INFO] [1637030892.112405885]: C : 16, sum : 746, Cumulative Average : 21.31
[ INFO] [1637030892.212443713]: C : 20, sum : 766, Cumulative Average : 21.28
[ INFO] [1637030892.312407017]: C : 14, sum : 780, Cumulative Average : 21.08
[ INFO] [1637030892.412424407]: C : 21, sum : 801, Cumulative Average : 21.08
```