**Opis funkcji aplikacji :**

1. Serwer - Oczekuje w nieskończonej pętli na żądanie od klienta/klientów, gdy odbierze prośbę o owoc(banana, orange). Wysyła wiadomość zwrotną o treści uzależnionej od stanu czapki sprzedawcy. Gniazdo serwera jest ustawione na pasywne:
   1. Struktury
      1. **CLIENT\_INFO** - struktura odpowiedzialna za przechowywanie informacji o kliencie oraz o tym czy sprzedawca ma czapkę bądź nie .
         1. Zmienne w strukturze:
            1. int clientId - przechowuje id klienta.
            2. SOCKET clientSocket - przechowuje gniazdo przez które klient nawiązuje połączenie z serwerem.
            3. char\* ip - przechowuje ip klienta.
            4. BOOLEAN hat - przechowuje informacje o stanie czapki u sprzedawcy.
            5. CRITICAL\_SECTION cr - przechowuje informacje o sekcji krytycznej.

* 1. Funkcje
     1. **int main(void)** **-** inicjalizujemy naszego winsocka, Gniazdo klienta oraz Gniazdo nasłuchowe. W pętli while nasłuchujemy próśb od poszczególnych klientów.Oraz tworzymy dla każdego klienta osobny wątek.
     2. **DWORD WINAPI MessageHandler(void\* sd\_)** - funkcja wywoływana przez wątek kliencki.
     3. **BOOLEAN MessageIncomingPackets(void \*params)-**funkcja odpowiedzialna za odbiór informacji od klienta. Oraz odpowiedź zwrotną do klienta o treści uzależnionej od stanu czapki sprzedawcy.

Funkcja również modyfikuje stan czapki sprzedawcy a jako że ta informacja jest wspólna dla wszystkich wątków klienckich w celu uniknięcia błędów spowodowanych jednoczesnym dostępem i modyfikacją stanu czapki. Konieczne stało się zainicjalizowanie współdzielonego obszaru pamięci za pomocą sekcji krytycznej.

2. Klient - Wysyła do serwera jedno z dwóch poleceń (banana albo orange). Po czym nasłuchuje w nieskończonej pętli odpowiedzi od serwera. Wiadomość zwrotna uzależniona jest od treści wysłanego polecenia Gniazdo klienta ustawione jest na aktywne.