

PSIR — Projekt

Michał Drętkiewicz

Eryk Głąb

Patryk Kosiński

13.01.2024

Spis treści

1 Specyfikacja protokołu aplikacyjnego	3
2 Tuple Space	4
3 Serwer	4
4 Aplikacja 1.	4
5 Aplikacja 2.	4
6 Environment file dla aplikacji 2.	4

1 Specyfikacja protokołu aplikacyjnego

Implementacja protokołu zawarta jest w plikach `protocol.h` i `protocol.h`.

Wiadomości protokołu mają następującą strukturę:

```
typedef struct Message {
    uint32_t id;
    MessageType type;
    MessageData data;
} Message;
```

Znaczenie pól:

- **id** – jest to unikalny w zakresie każdego połączenia identyfikator wiadomości, wykorzystywany głównie w mechanizmie potwierdzania odbioru (ACK). Kolejne wartości są generowane funkcją `message_next_id()`.
- **type** – rodzaj wiadomości. Definicja typu `MessageType`:

```
typedef enum MessageType {
    message_ack,
    message_tuple_space_insert_request,
    message_tuple_space_get_request,
    message_tuple_space_get_reply,
} MessageType;
```

Znaczenie wartości:

- **message_ack** – potwierdzenie odebrania wcześniejszej wiadomości, przesyłane w obu kierunkach,
 - **message_tuple_space_insert_request** – żądanie wstawienia krotki do przestrzeni krotek, wysyłane od klienta do serwera,
 - **message_tuple_space_get_request** – żądanie odczytania krotki z przestrzeni krotek, wysyłane od klienta do serwera,
 - **message_tuple_space_get_reply** – odpowiedź na wiadomość typu `message_tuple_space_get_request`, przesyłana od serwera do klienta.
- **data** – reszta danych wiadomości, zależna od jej typu. Definicja typu `MessageData`:

```
typedef union MessageData {
    MessageAck ack;
    MessageTupleSpaceInsertRequest tuple_space_insert_request;
    MessageTupleSpaceGetRequest tuple_space_get_request;
    MessageTupleSpaceGetReply tuple_space_get_reply;
} MessageData;
```

Poszczególne pola unii odpowiadają kolejno wartościom `MessageType` opisanym w poprzednim punkcie.

Definicje typów pól:

- ```
typedef struct MessageAck {
 uint32_t message_id;
} MessageAck;
```

Pole **message\_id** to id wiadomości, której dotyczy to potwierdzenie.

-

```
typedef struct MessageTupleSpaceInsertRequest {
 Tuple tuple;
} MessageTupleSpaceInsertRequest;
```

Znaczenie pola takie samo, jak w argumentach funkcji `tuple_space_insert()`, opisanej w następnym rozdziale.

- ```
typedef struct MessageTupleSpaceGetRequest {
    Tuple tuple_template;
    TupleSpaceOperationBlockingMode blocking_mode;
    TupleSpaceOperationRemovePolicy remove_policy;
} MessageTupleSpaceGetRequest;
```

Znaczenie pól takie samo, jak w argumentach funkcji `tuple_space_get()`, opisanej w następnym rozdziale.

```
typedef struct MessageTupleSpaceGetReply {
    TupleSpaceOperationResult result;
} MessageTupleSpaceGetReply;
```

Znaczenie pola takie samo, jak w wartości zwracanej z funkcji `tuple_space_get()`, opisanej w następnym rozdziale.

Podczas transmisji, każda wiadomość poprzedzona jest swoją długością w bajtach, rzutowaną na typ `uint32_t`.

2 Tuple Space

3 Serwer

4 Aplikacja 1.

5 Aplikacja 2.

6 Environment file dla aplikacji 2.