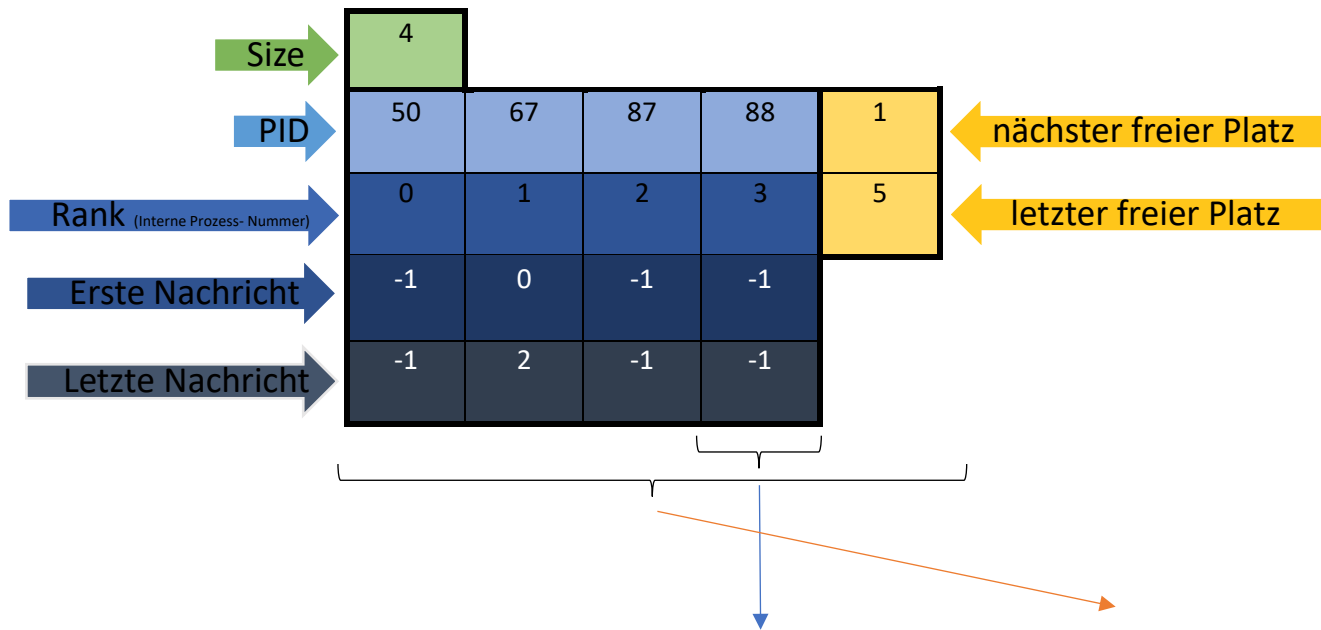


Datenstruktur des Shared Memory



```
typedef struct {
    int first;
    int last;
} FIRST_LAST;
```

```
typedef struct {
    pid_t process_id;
    FIRST_LAST first_last_msg;
} PROCESS_HEADER;
```

```
typedef struct {
    unsigned int size;
    FIRST_LAST first_last_free;
    PROCESS_HEADER processes[];
} HEADER;
```

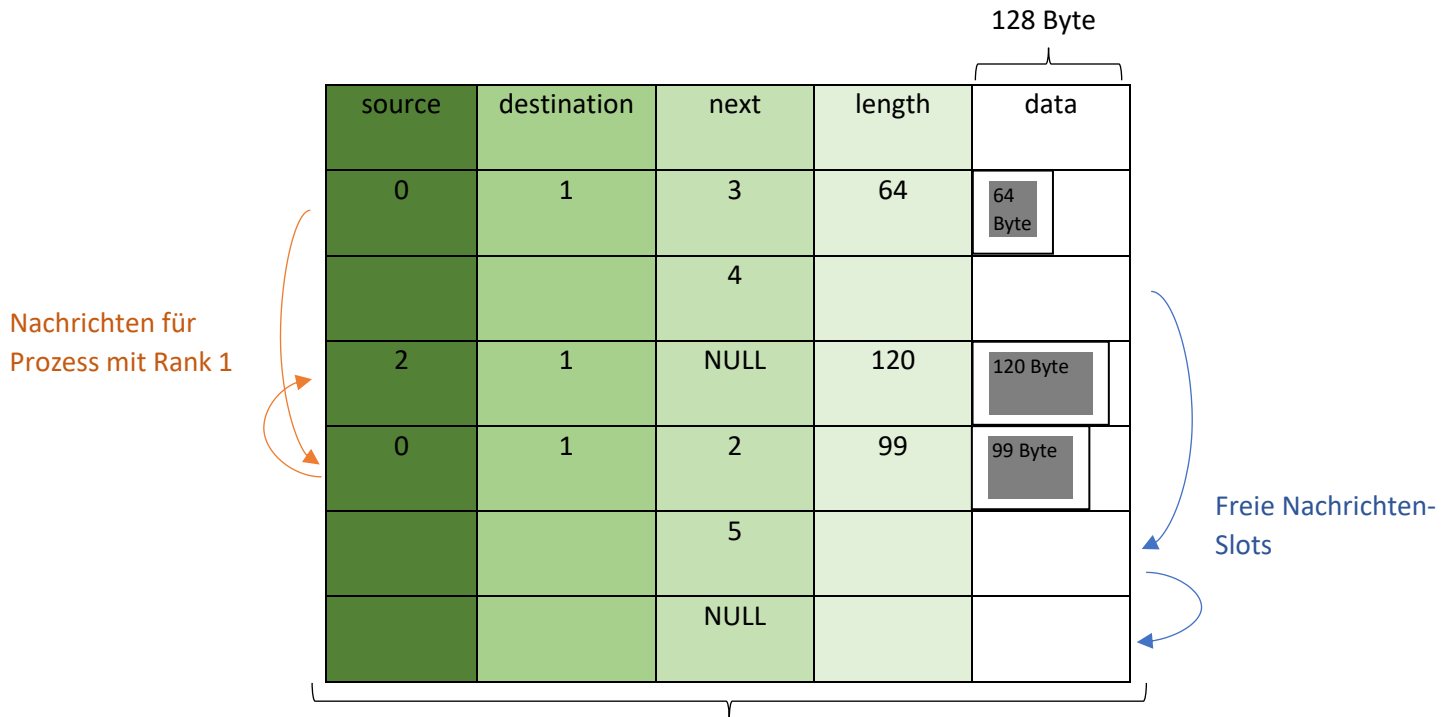
Erläuterungen:

process_id: Die jeweilige Prozess- Identifikations- Nummer des jeweiligen Kind- Prozesses.

first_last_msg: Die eingetragene Zahl entspricht dem jeweiligen Nachrichten- Slot, wo die erste Nachricht bzw. die letzte Nachricht an den jeweiligen Prozess gespeichert ist.

Size: Die gesamte Anzahl aller Kindprozesse.

first_last_free: Die eingetragene Zahl entspricht dem jeweiligen ersten bzw. letzten Nachrichten- Slot.



```
typedef struct {
    unsigned int destination;
    unsigned int source;
    int next;
    size_t length;
    char data[OSMP_MAX_PAYLOAD_LENGTH];
} MESSAGE;
```

Erläuterungen:

destination: Angabe, an welchen Prozess die jeweilige Nachricht gerichtet ist.

source: Angabe, von welchem Prozess die jeweilige Nachricht stammt.

next: Je nach Situation steht der Inhalt des Feldes für eine Art Zeiger auf das nächste Feld mit der nächsten Nachricht für denselben Prozess bzw. auf das nächste Feld des nächsten freien Nachrichten- Slots.

data: Hier wird der Inhalt einer Nachricht mit einer fest definierten Länge (**length**) für den jeweiligen Prozess gespeichert.

```
typedef struct {
    MESSAGE messages[OSMP_MAX_SLOTS];
    HEADER header;
} SHARED_MEMORY;
```

Erläuterungen:

Die Datenstruktur für den Shared Memory wird nun aus einer festgelegten maximalen Anzahl an Nachrichten- Slots (OSMP_MAX_SLOTS) und aus dem Header definiert.