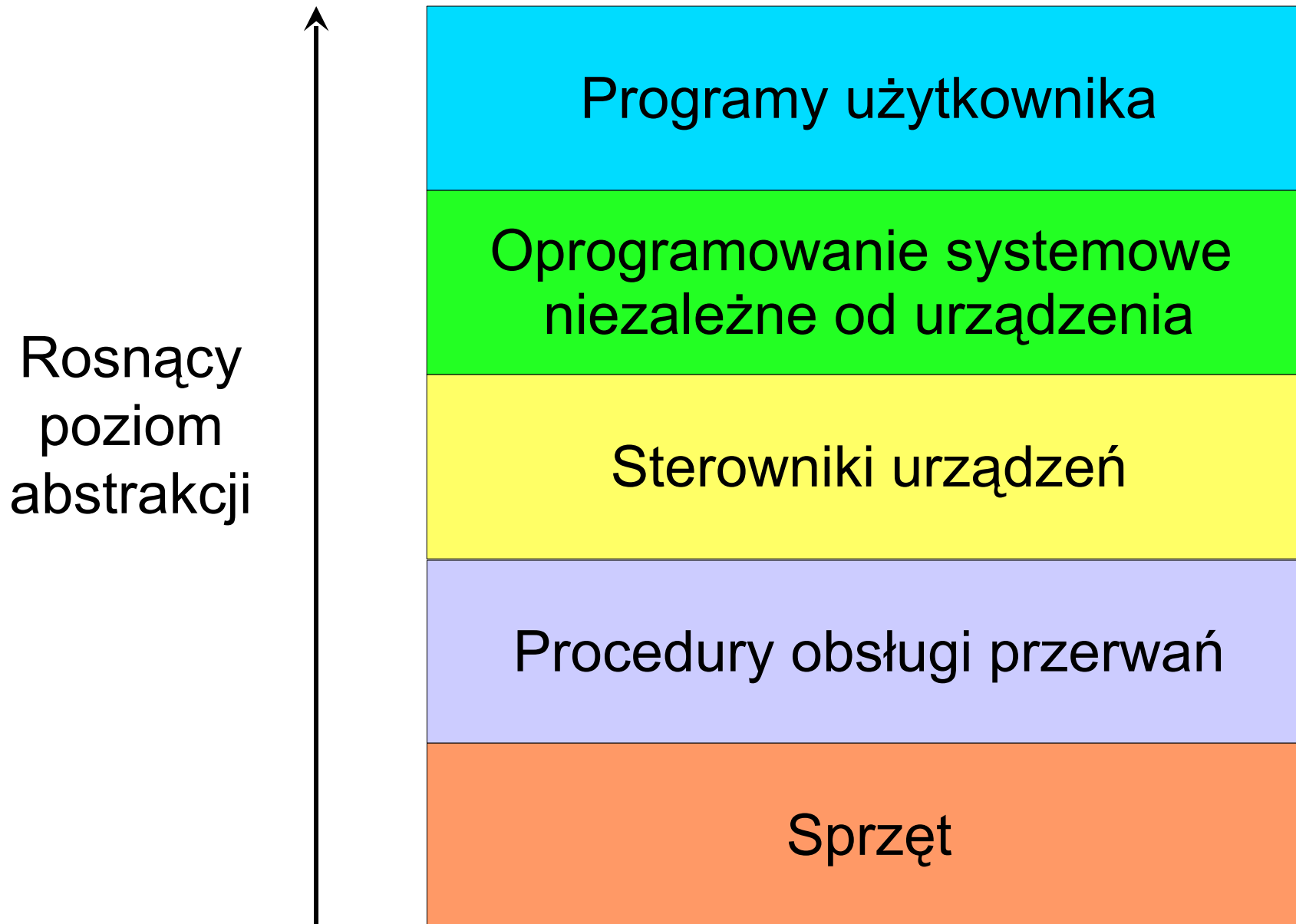


Warstwy oprogramowania wejścia/wyjścia



Funkcje sterownika urządzenia

- Jądro systemu komunikuje się ze sterownikiem poprzez wywołanie następujących operacji przezeń zaimplementowanych:
 - **init** - inicjalizacja sterownika
 - **open** - otwarcie urządzenia
 - **close** - zamknięcie urządzenia
 - **read** - zapis do urządzenia
 - **write** - odczyt z urządzenia
 - **ioctl** - różne funkcje sterujące

Funkcje sterownika urządzenia

- Aby jądro "wiedziało" o sterowniku urządzenia, należy informację o nim wpisać do tablicy sterowników

```
typedef rtems_device_driver ( *rtems_device_driver_entry ) (
    rtems_device_major_number,
    rtems_device_minor_number,
    void * );

typedef struct {
    rtems_device_driver_entry initialization_entry; /* initialization procedure */
    rtems_device_driver_entry open_entry; /* open request procedure */
    rtems_device_driver_entry close_entry; /* close request procedure */
    rtems_device_driver_entry read_entry; /* read request procedure */
    rtems_device_driver_entry write_entry; /* write request procedure */
    rtems_device_driver_entry control_entry; /* special functions procedure */
} rtems_driver_address_table;

#define DRV_DRIVER_TABLE_ENTRY \
    {drv_init, drv_open, drv_close, drv_read, drv_write, drv_ioctl}
#ifdef CONFIGURE_INIT
rtems_driver_address_table Device_drivers[2] =
{
    CONSOLE_DRIVER_TABLE_ENTRY,
    DRV_DRIVER_TABLE_ENTRY
};
#endif
```

Inicjalizacja urządzenia

```
rtems_device_driver
```

```
drv_init(rtems_device_major_number major,  
rtems_device_minor_number minor, void *  
context);
```

- Funkcja wywoływana automatycznie przez system operacyjny na etapie startu.
- Umożliwia przeprowadzenie konfiguracji urządzenia oraz wszystkich zmiennych sterownika.
- Pierwszy i drugi parametr funkcji identyfikuje urządzenie do którego jest skierowane polecenie. Trzeci parametr funkcji nie jest wykorzystywany.
- Ważnym zadaniem tej funkcji jest przypisanie nazwy urządzenia do numerów identyfikujących urządzenie. Zadanie to realizowane jest za pomocą funkcji `rtems_io_register_name`.

Otwarcie/zamknięcie urządzenia

- Funkcje odpowiedzialne za inicjalizację sterownika w momencie próby otworzenia dostępu do urządzenia oraz zamknięcie dostępu do urządzenia.
- Umożliwia między innymi kontrolę poprawności praw dostępu do urządzenia. Trzeci parametr funkcji jest wskaźnikiem do struktury `rtems_libio_open_close_args_t`
- Struktura zdefiniowana i skomentowana w `rtems/libio.h`

```
typedef struct {
    rtems_libio_t *iop;
    uint32_t flags /* opening mode */;
    uint32_t mode /* access permissions */;
} rtems_libio_open_close_args_t;
```

Odczyt/zapis do/z urządzenia

- Trzeci parametr funkcji jest wskaźnikiem do struktury `rtems_libio_rw_args_t`

```
typedef struct {
    rtems_libio_t *iop;
    off_t offset;
    char *buffer;
    uint32_t count;
    uint32_t flags;
    uint32_t bytes_moved;
} rtems_libio_rw_args_t;
```

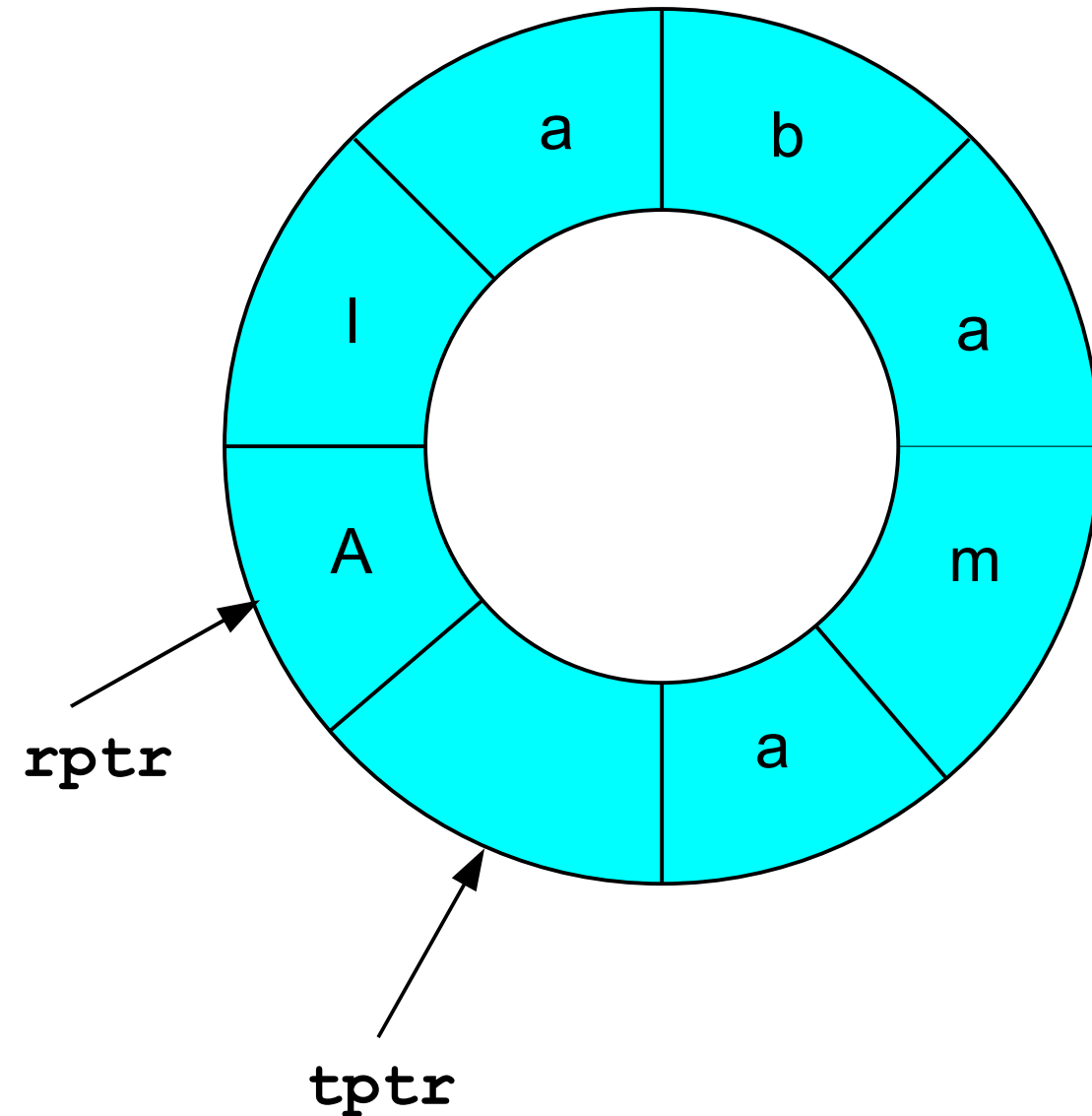
Pozostałe operacje - ioctl

- Trzeci parametr funkcji jest wskaźnikiem do struktury `rtems_libio_ioctl_args_t`

```
typedef struct {
    rtems_libio_t *iop;
    uint32_t command;
    void *buffer;
    uint32_t ioctl_return;
} rtems_libio_ioctl_args_t;
```

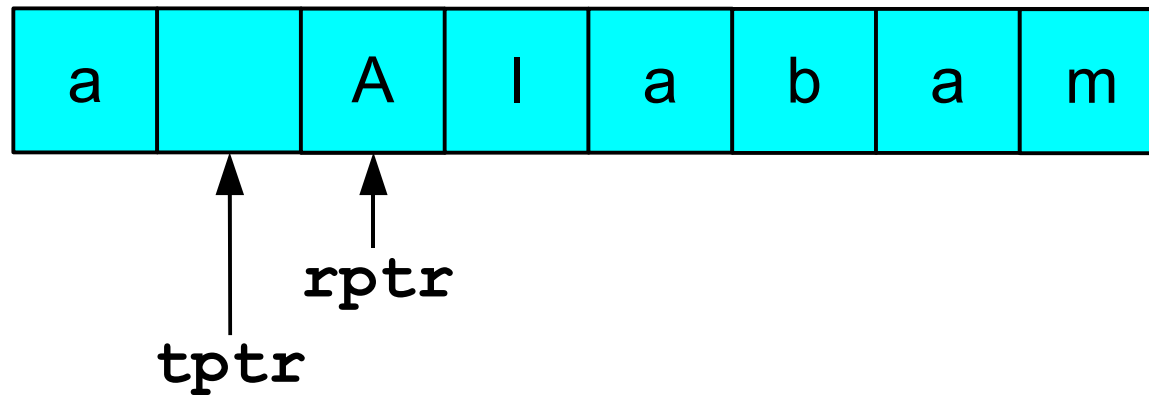
- Identyfikator wykonywanego polecenia jest liczbą 32 bitową składającą się z czterech ośmiobitowych części określających poszczególne właściwości danego polecenia:
 - Liczba magiczna - ośmiobitowa liczba wybierana przez programistę
 - Kolejna liczba ośmiobitowa
 - Kierunek przesyłania danych (8 bitów):
 - IOC NONE - brak przesyłania danych
 - IOC READ - odczyt danych z urządzenia
 - IOC WRITE - zapis danych do urządzenia
 - Ilość przesyłanych danych

Bufor kołowy



- problem producenta i konsumenta
- dane przekazywane i odbierane asynchronicznie
- konieczne buforowanie
- **rp** - receive pointer
- **tp** - transmit pointer
- piszemy pod **tp**, czytamy z **rp**
- **rp == tp** - bufor pusty
- **tp** zaraz przed **rp** - bufor pełny
- jedno miejsce w buforze zawsze się marnuje

Bufor kołowy



- W praktycznych implementacjach bufor jest liniowy, a arytmetyka na indeksach odbywa się modulo rozmiar bufora
- `rptr == tptr` - bufor pusty
- `(tptr + 1) % 8 == rptr` - bufor pełny

Zarządzanie buforem przy pomocy semaforów

