



Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
Wydział Elektrotechniki, Automatyki,
Informatyki i Inżynierii Biomedycznej

Programowanie Współbieżne i Rozproszone
Kierunek Informatyka
III rok
2020/2021

Maryna Lukachyk,
Yauheniya Padbiarozskaya
grupa 5a

Pakiet do obsługi urządzeń zewnętrznych
USB

20.01.2021

Cel programu:

Stworzenie programu symulującego obsługę urządzeń zewnętrznych przez port USB. Program został napisany w języku Erlang.

Opis i schemat struktury programu:

- *usb.erl* – start symulacji, główne menu obsługi portów USB, funkcje realizujące obsługę portów (dodawanie/usuwanie urządzenia, przegląd, obsługa, wyjście z symulacji).
- *filesUtils.erl* – funkcje potrzebne do zarządzania zawartością wybranego USB (dodawanie/usuwanie plików, przegląd).
- *utils.erl* – drugorzędne funkcje do wyświetlania danych wyjściowych.

>>> usb.erl

Program uruchamiany poprzez procedurę *start()*, która tworzy główny proces wyświetlający menu i wywołujący funkcje wybrane przez użytkownika.

Szereg działań dostępnych w symulacji dostępny przez główne menu:

- przegląd podłączonych urządzeń
- dodanie nowego urządzenia USB
- obsługa podłączonych urządzeń
- odłączenie urządzenia
- i oczywiście, wyjście z programu

show_connected_devices()

sprawdza długość listy urządzeń przekazanej do funkcji: jeśli lista pusta, wypisuje komunikat o braku podłączonych urządzeń, w przeciwnym razie wypisuje nazwy zawierające się w liście.

add_device()

tworzy nowy proces, który wysyła komunikat do funkcji dodającej nowe urządzenia *add_device_function()*, ta funkcja z kolei odbiera sygnał, aktualizuje listę przechowującą wszystkie urządzenia, a potem wysyła z powrotem komunikat, który sygnalizuje, że pomyślnie dodano nowe urządzenie, zwiększ liczbę urządzeń o 1.

manage_devices()

wypisuje listę urządzeń, z której użytkownik wybiera pozycję do obsługi. Program sprawdza listę urządzeń i szuka identyfikatoru, który odpowiada indeksu wprowadzonemu przez użytkownika do konsoli. Jeśli długość listy znalezionych PID'ów jest różna od zera, wysyła do jej pierwszego elementu sygnał w postaci {DevicePID, menu} – wyświetl menu dostępnych konfiguracji urządzenia.

disconnect()

sprawdza długość listy urządzeń, jeśli różni się od zera szuka PID w liście, który odpowiada indeksu wprowadzonemu przez użytkownika do konsoli. Do elementu posiadającego PID wysyła sygnał w postaci {kill}, aktualizuje listę urządzeń poprzez wywołanie *updateDevicesMap()*.

...

>>> *filesUtils.erl*

remove_file() i *add_file()*

Potrzebują listę plików na urządzeniu. *Add_file()* pobiera od użytkownika nazwę i rozszerzenie pliku, które przekazuje do funkcji *addFiles()*. *Remove_file()* szuka ID pliku w liście plików, po znalezieniu wywołuje *removeFile()*, menu obsługi urządzenia z aktualną listą plików na urządzeniu.

...

Pakiety zewnętrzne:

Instrukcja obsługi:

Ograniczenia programu: