

Opis architektury hosta:

Cały host (a tak naprawdę cały symulator płytki wraz GUI) został zrealizowany przy użyciu języka Java. Komunikacja między hostem a płytką została zrealizowana za pomocą socketu(dwukierunkowy mechanizm komunikacyjny), gdzie Host pełni rolę klienta(klasa Socket w Javie), a Board(nazwa klasy płytki) pełni rolę serwera(klasa ServerSocket w Javie). Oba elementy zostają na sztywno przypisane do portu 9999. Wysyłanie wiadomości na płytkę zostało zrealizowane za pomocą klasy BufferedReader, a odbieranie wiadomości za pomocą PrintStream. Wiadomości zarówno wysyłane na płytkę, jak i odbierane z płytki mają rozmiar 8. Po zaakceptowaniu połączenia przez klasę Board Host przechodzi do dalszego działania. Funkcjonalności Hosta zostały podzielone na cztery elementy:

1) Ustawianie na PC poziomu audio za pomocą potencjometru na płytce oraz wyświetlanie diod LED na płytce w zależności od poziomu audio na PC – podpunkty 1 i 2 z oryginalnego projektu zostały połączone.

2) Wyświetlanie obciążenia systemu hosta na wyświetlaczu HD44780

3) Przyciski S1-S8 uruchamiające dowolną akcję na hoście oraz wyświetlanie ich opisu w reakcji na kliknięcie – podpunkty 4 i 5 z oryginalnego projektu zostały połączone

4) Wyświetlanie na diodzie RGB bieżącego uśrednionego koloru strumienia wideo

Po wybraniu przez użytkownika funkcjonalności host wysyła na płytkę wiadomość, która informuje o realizacji danego elementu systemu.

Każdy z poniższych elementów kończy działanie po odebraniu sygnału SIGNINT(CTRL) lub po wyłączeniu płytki(wyłapanie wyjątku SocketException).

1) Ustawianie na PC poziomu audio za pomocą potencjometru na płytce oraz wyświetlanie diod LED na płytce w zależności od poziomu audio na PC – Wiktor Sokół, Maciek Szybecki, Łukasz Stolarz, Michał Sz.

Host zaczyna odczytywać w pętli wiadomości nadawane z płytki, dopóki nie są one puste. Dla przesłanej wartości userInput z poziomu Javy wywoływana jest proces realizujący komendę linuxową „amixer”, która ustawia poziom audio. Dodatkowo bieżący wątek zostaje zablokowany do momentu ustawienia dźwięku. Potem ponownie za pomocą komendy linuxowej „amixer” oraz ProcessBuilder-a wartość audio zostaje przekazana do Stringa i odpowiednio przycięta. Bieżący wątek ponownie zostaje zablokowany do momentu wykonania procesu. Tak pozyskana wartość zostaje przekazana na płytkę.

2) Wyświetlanie obciążenia systemu hosta na wyświetlaczu HD44780 – Jakub Stawowy, Grzegorz Szydło

Dla obiektu Runtime związanego z bieżącą aplikacją zostają wywołane następujące komendy linuksowe:

```
cat /sys/class/thermal/thermal_zone0/temp  
free | grep Mem | cut -d \" \" -f8  
free | grep Mem | cut -d \" \" -f13  
grep 'cpu ' /proc/stat | awk '{usage=($2+$4)*100/($2+$4+$5)} END {print usage}'
```

Dzięki temu pozyskujemy temperaturę CPU, % zużycie RAM oraz procentowe zużycie CPU (dla RAM-u wartość 2 zostaje podzielona przez wartość 3). Wartości z nich pozyskane zostają zapisane do odpowiednich String-ów za pośrednictwem obiektów `BufferedReader`, a bieżący wątek jest blokowany do momentu wykonania wszystkich komend. Parametry obciążenia systemu zostają wysłane w odpowiednich odstępach czasowych.

3) Przyciski S1-S8 uruchamiające dowolną akcję na hoście oraz wyświetlanie ich opisu w reakcji na kliknięcie- Paweł Sikora, Mateusz Sitek, Paweł Szydło, Bartosz Szlęzak

Host odbiera od płytki wiadomość o treści od 1 do 8. W zależności od tego na podstawie instrukcji warunkowej zostaje wywołane różne strony internetowe. Następnie na płytkę zostaje wysłana wiadomość o dokładnie takiej samej treści, która posłuży płytce do wywołania przechowywanego opisu.

4) Wyświetlanie na diodzie RGB bieżącego uśrednionego koloru strumienia wideo – Jakub Medalion, Mateusz Rychlik

Za pomocą „domyślnego zestawu narzędzi” zostają pobrane i przypisane do wartości `width` i `height` szerokość ekranu i wysokość ekranu. Następnie zostaje przechwycony prostokąt o wymiarach `width` i `height`, czyli faktycznie cały ekran. Dla tak przechwyconego obrazu dla pikseli z siatki o odstępach w pikselach wyznaczonym przez parametr `SECT_SKIP` zostają pobrane wartości: Red, Green, Blue, a następnie ich suma jest dzielona przez ilość pikseli w siatce. Na końcu host wysyła je na płytkę.