

1. Opis działania projektowanego systemu

Projektowany system służy do przetwarzania dźwięku w czasie rzeczywistym i dodawania do niego efektu echa. Dźwięk jest pobierany z wejścia karty dźwiękowej (mikrofonu). Na podstawie parametrów zadanych przez użytkownika takich, jak liczba ścieżek echa i ich głośność, przesunięcie w czasie każdej ścieżki względem poprzedniej (czas między rozpoczęciem każdej z nich) oraz stałe opóźnienie między momentem wprowadzenia dźwięku a chwilą jego wyprowadzenia przez system, program przetwarza fragmenty dźwięku otrzymane z wejścia i ma za zadanie przekazać przetworzony dźwięk na wyjście karty dźwiękowej (do głośników) z zadaniem wcześniej, stałym opóźnieniem. Ograniczeniem czasu rzeczywistego w naszym systemie jest czas między momentem wprowadzenia a wyprowadzenia dźwięku z systemu. Naszym celem jest to, aby ten czas wynosił dokładnie tyle, ile podał użytkownik i aby system spełniał wymagania czasu rzeczywistego dla jak najkrótszych czasów podanych przez użytkownika.

Prezentujemy uproszczony sposób dodawania echa. P_x to kolejna próbka dźwięku, który przetwarzamy, P_x' to przetworzona już próbka (z dodanym echem), Δt to przesunięcie poszczególnych ścieżek audio, a α_x to ułamek reprezentujący poziom głośności poszczególnych ścieżek. Poniżej znajduje się schemat, który pokazuje, jak to wygląda oraz sposób obliczania poszczególnych próbek. Innymi słowy, nakładamy na siebie konkretne próbki z odpowiednią głośnością.

| | P_1' | P_2' | P_3' | P_4' | P_5' | P_6' | P_7' | P_8' | P_9' |
|------------|-------------|--------|--------|--------|--------|--------|--------|--------|--------|
| α_0 | P_1 | P_2 | P_3 | P_4 | P_5 | P_6 | P_7 | P_8 | P_9 |
| α_1 | Δt | | P_1 | P_2 | P_3 | P_4 | P_5 | P_6 | P_7 |
| α_2 | $2\Delta t$ | | | | P_1 | P_2 | P_3 | P_4 | P_5 |

$$P_1' = \alpha_0 * P_1$$

$$P_2' = \alpha_0 * P_2$$

$$P_3' = \alpha_0 * P_3 + \alpha_1 * P_1$$

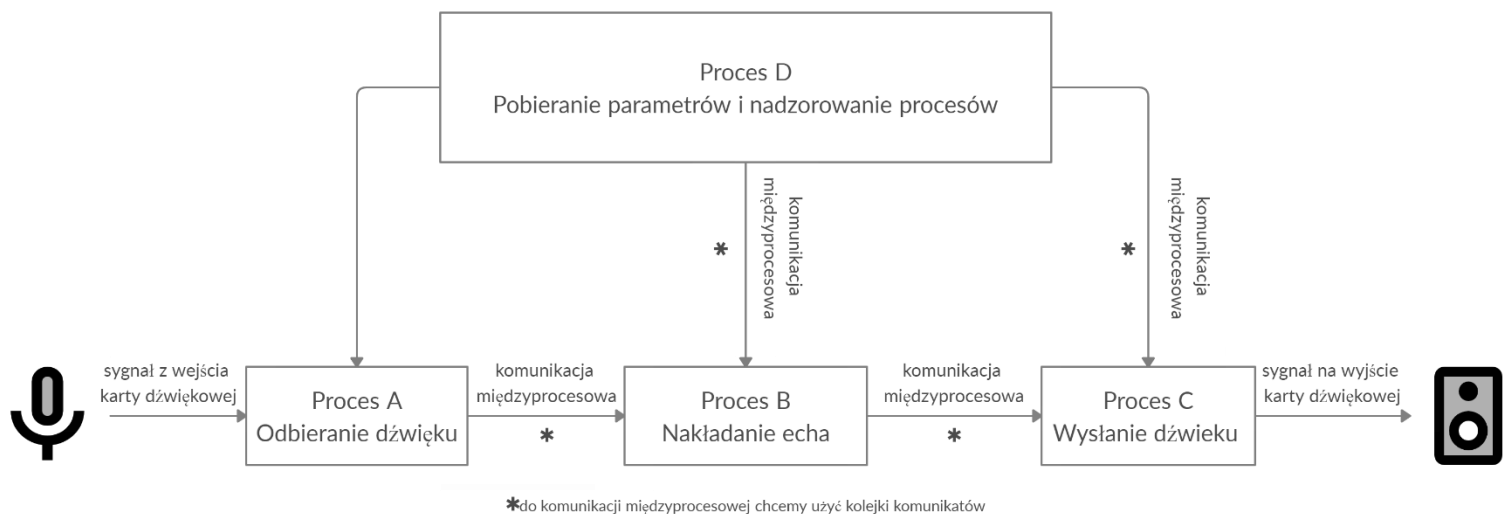
$$P_4' = \alpha_0 * P_4 + \alpha_1 * P_2$$

$$P_5' = \alpha_0 * P_5 + \alpha_1 * P_3 + \alpha_2 * P_1$$

$$P_6' = \alpha_0 * P_6 + \alpha_1 * P_4 + \alpha_2 * P_2$$

Program będziemy realizować na systemie operacyjnym Ubuntu Lite uruchamianym z bootowalnego pendrive'a. Do obsługi dźwięku wykorzystamy bibliotekę PortAudio (wybór biblioteki jest wstępny). Strojenie systemu zrealizujemy przez optymalizację kodu, ustawianie priorytetów naszych procesów w systemie oraz odpowiednie zależności między procesami.

2. Schemat systemu



3. Opis poszczególnych elementów systemu

- Proces A - pobieranie dźwięku:
Proces A pobiera z wejścia karty dźwiękowej (mikrofonu) sygnał dźwiękowy ze stałą częstotliwością, przetwarza go na bloki o zadanej długości i przekazuje procesowi B (przez kolejkę komunikatów) do obróbki.
- Proces B - przetwarzanie dźwięku:
Proces B otrzymuje od procesu A bloki danych dźwięku i, zgodnie z opisany wcześniej sposobem, nakłada echo na podstawie parametrów zadanych przez użytkownika (otrzymanych od procesu D). Tak przetworzony dźwięk przekazuje (przez kolejkę komunikatów) do procesu C.
- Proces C - wysłanie dźwięku:
Proces C odbiera przetworzone przez proces B dane i, zgodnie z zadany przez użytkownika opóźnieniem (otrzymanym od procesu D), podaje sygnał dźwiękowy na wyjście karty dźwiękowej.
- Proces D - zarządzanie parametrami pracy systemu:
Proces D inicjalizuje wszystkie pozostałe procesy oraz realizuje interfejs użytkownika: można tam ustawić liczbę ścieżek echa i ich głośność, przesunięcie między ścieżkami oraz opóźnienie, z jakim dźwięk podany do systemu jest przepuszczany na wyjście karty dźwiękowej. Zmiany ustawienia dotyczące opóźnienia między wprowadzeniem a wyprowadzeniem dźwięku są wysyłane do procesu C przez kolejkę komunikatów, podobnie ze zmianami w opisie echa - są wysyłane do procesu B przez inną kolejkę komunikatów.
- Komunikacja międzyprocesowa:
Komunikacja międzyprocesowa jest realizowana przy użyciu systemowych kolejek komunikatów w systemie typu Linux. Rozważaliśmy też pamięć współdzieloną, lecz nasz wybór padł na kolejkę komunikatów, ponieważ w naszym systemie konkretny rodzaj danych jest wysyłany tylko z jednego określonego procesu do drugiego określonego procesu, zatem nie musimy realizować synchronizacji dostępu do pamięci współdzielonej.

4. Eksperymenty możliwe do przeprowadzenia

- Badanie wpływu sposobu szeregowania poszczególnych procesów na to, czy program jest w stanie spełnić postawione mu wymagania czasu rzeczywistego (próbujemy szeregować procesy na różne sposoby i sprawdzamy wartości opóźnienia między przekazaniem dźwięku na wejście a wyprowadzeniem echa – porównujemy zadane opóźnienia z uzyskanymi pomiarami czasów i przedstawiamy je na wykresie punktowym. Wyniki prezentujemy w tabelach i tworzymy wykresy dla różnych rodzajów szeregowania i patrzymy, jak znacząco wyniki odbiegają od zadanych opóźnień).
- Badanie wpływu liczby nakładanych warstw echa oraz przesunięcia w czasie każdej ścieżki względem poprzedniej na złożoność obliczeń, a tym samym na spełnianie wymagań czasu rzeczywistego – podobnie, jak w poprzednim punkcie, porównujemy wartości zadanego opóźnienia z tym, co uzyskujemy w rzeczywistości, a ponadto dokonujemy pomiarów czasów obróbki dźwięku w zależności od liczby warstw echa i przesunięcia w czasie ścieżek (tabele oraz wykres punktowy, jak w poprzednim punkcie oraz drugi prezentujący, jak zmienia się czas przetwarzania dźwięku w zależności od złożoności obliczeń).
- Badanie, jakie najkrótsze wartości opóźnienia między chwilą pobrania dźwięku z wejścia, a wyprowadzeniem przetworzonego dźwięku na wyjście da się osiągnąć i jakie działania na to pozwalają na podstawie innych przeprowadzonych badań oraz przedstawienie wyników i wniosków w formie opisowej.
- Badanie wpływu częstotliwości pobierania danych z wejścia na działanie systemu (analogicznie do punktu drugiego).
- Badanie działania programu w zależności od stopnia obciążenia systemu operacyjnego (próbujemy stopniowo obciążać system operacyjny, np. poprzez uruchamianie kolejnych procesów i dokonujemy porównania czasów opóźnienia między wejściem dźwięku a wyjściem echa z naszego systemu, wyniki prezentujemy w tabelach oraz na wykresie punktowym).

5. Podział prac w zespole

- **Michał Kopyt:**
Szkielet programu
Utworzenie procesu D
Eksperymenty
- **Paweł Martyniuk:**
Utworzenie Procesu A
Utworzenie Procesu C
Komunikacja międzyprocesowa (kolejki komunikatów)
- **Adrian Prorok:**
Szkielet programu
Utworzenie Procesu B
Eksperymenty