

Multimedialne interfejsy programistyczne API

1

Paweł Spaleniak

Katedra Systemów Multimedialnych

- Ogólne wiadomości na temat rejestracji, przetwarzania i odtwarzania multimedialnych
- Platforma Windows
- Platforma Linux
- Biblioteki wieloplatformowe
- Platforma JAVA

- Pojedyncza próbka dźwięku ma zbyt dużą ziarnistość, aby przetwarzanie „próbka-po-próbce” było efektywne
- Próbki dźwięku łączone są w bloki, które są podstawowym „kwantem” danych przy pracy z dźwiękiem
- Typowe parametry bloku (bufora):
 - Ilość kanałów dźwięku
 - Długość bloku (ilość próbek należących do 1 kanału)
 - Ilość bitów opisujących pojedynczą próbkę
 - Konfiguracja (przeplot) kanałów
 - Wyrównanie (alignment) próbek - niektóre architektury procesorów wymagają np. żeby adresy w pamięci poszczególnych próbek były podzielne przez 4.

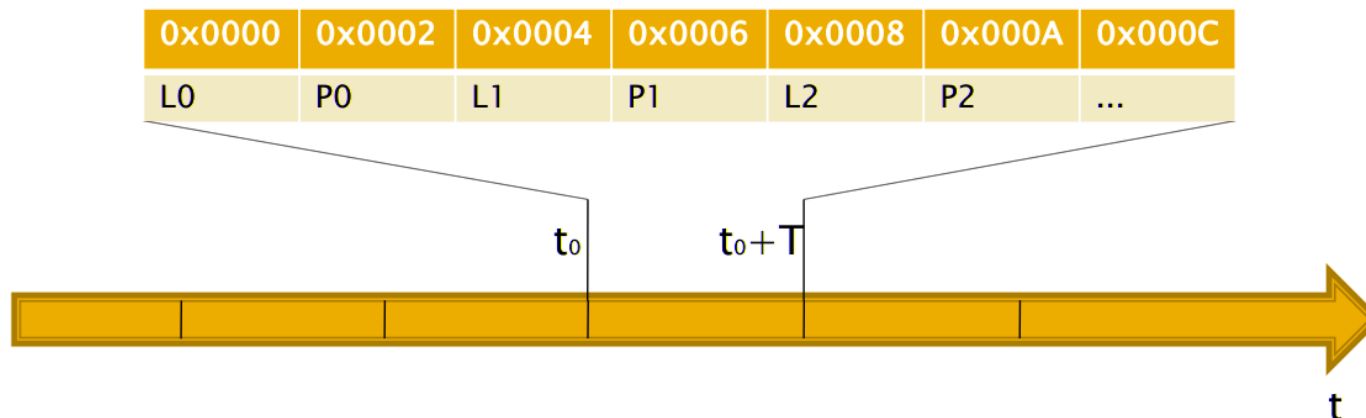
Rejestracja dźwięku - buforowanie

4

- Konsekwencją buforowania jest **bezwładność** (latency) - opóźnienie proporcjonalne do długości bufora
- Zastosowanie krótkich buforów pozwala uzyskać niską bezwładność, kosztem większego użycia czasu procesora

BUFOR

- 2 kanały (L+P), 16bit/próbkę, przeplot LPLP, wyrównanie na granicy 2-bajtów



FORMATY STAŁOPOZYCYJNE (całkowite)

- **U8** - 8-bitowe liczby całkowite bez znaku, zakres 0-255 (256 poziomów kwantyzacji), dynamika ~48dB; 1 próbka = 1B; 0 (cisza cyfrowa) reprezentowana przez 128 (0x80)
- **I16** - 16-bitowe liczby całkowite ze znakiem (format płyt CD), zakres -32768 - 32767; dynamika ~96dB; 1 próbka = 2B
- **I24** - 24-bitowe liczby całkowite ze znakiem; dynamika ~144dB; format występuje w postaci „spakowanej” (3B) i „rozpakowanej” (4B - padding)
- **I32** - 32-bitowe liczby całkowite ze znakiem, tożsamy z „rozpakowanym” formatem I24

FORMAT ZMIENNOPOZYCYJNY (F32)

- 32-bitowe liczby rzeczywiste (IEEE 754)
- zakres użyteczny wykorzystywany przez dane dźwiękowe - 1,0 - 1,0
- ze względu na wykorzystanie tylko „wycinka” zakresu liczb float, nie ma niebezpieczeństwa przesterowania cyfrowego podczas wykonywania obliczeń
- konieczność ograniczenia zakresu danych przed odtworzeniem ich

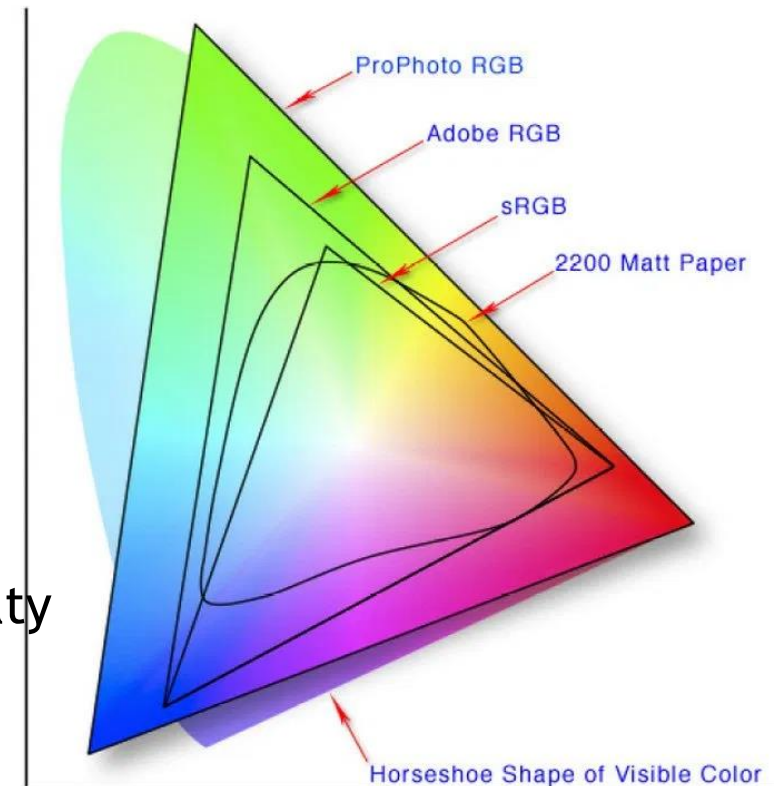
- **Kolejność bajtów** - architektury procesorów różnią się w sposobie reprezentacji w pamięci liczb wielobajtowych (byte ordering)

0x01	0x02	0x03	0x04
LSB (B0)	B1	B2	MSB (B3)
Little-endian (Intel, ARM)			

0x01	0x02	0x03	0x04
MSB (B3)	B2	B1	LSB (B0)
Big-endian (Motorola, IBM Power)			

- Niektóre formaty danych używają stałej kolejności bajtów, inne - zależnej od architektury procesora (konieczność konwersji formatów)

- Ramka obrazu stanowi „mapę bitową”, poszczególne bity kodują składowe koloru
- Kolor piksela wyrażany w postaci n-tki składowych (najczęściej 3 lub 3+alpha)
- Składowe koloru określa przestrzeń barw formatu wideo
- Najczęściej spotykane przestrzenie barw:
 - **RGB** (red, green, blue) - typowo wykorzystywana do wyświetlania obrazu na monitorze
 - **YCbCr/YUV** (luminancja + 2 składowe chrominancji) - formaty wykorzystywane w standardach MPEG itp.
 - Różne wariacje powyższych polegające na ograniczeniu zakresu próbek lub przeskalowania krzywą gamma



Rejestracja wideo - organizacja ramki w pamięci

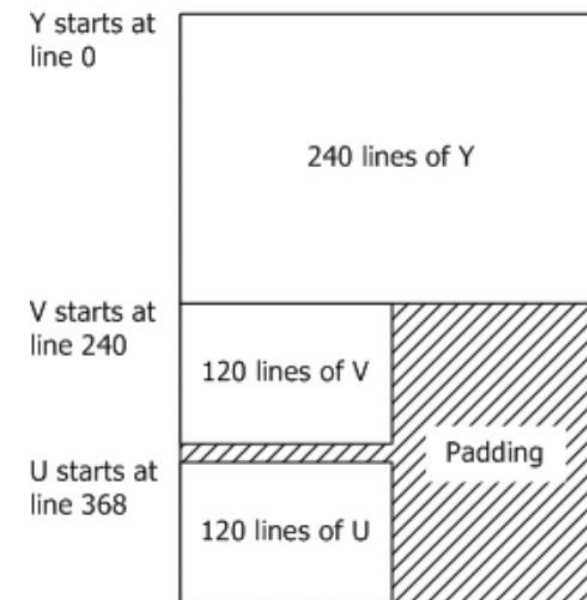
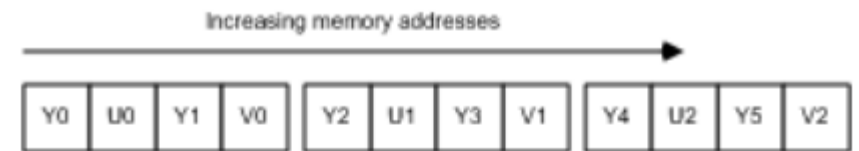
- Niektóre formaty oparte na przestrzeni barw YCbCr stosują subsampling składowych chrominancji - redukcja rozmiaru danych
- Organizacja rozmieszczenia danych pojedynczej składowej w pamięci może być wyrażona za pomocą 5 zmiennych:
 - **Width** - ilość pikseli w 1 linii obrazu
 - **Height** - ilość linii obrazu
 - **Start** - adres 0. piksela składowej względem początku bufora ramki
 - **LineStride** - przesunięcie pomiędzy adresami 0. piksela w kolejnych liniach
 - **PixelStride** - przesunięcie pomiędzy adresami kolejnych pikseli tej samej linii
- Adres piksela (x, y) w buforze danych wideo można określić wzorem:

$$\mathbf{Start + y * LineStride + x * PixelStride}$$

Rejestracja wideo - organizacja ramki w pamięci

10

- **YUY2 - YCbCr 4:2:2 z przeplotem**
 - Y: (320, 240, 0, 2 * 320, 2)
 - U: (320, 120, 1, 4 * 320, 4)
 - V: (320, 120, 3, 4 * 320, 4)
- **IMC1 - YCbCr 4:2:0 planarny z paddingiem**
 - Y: (320, 240, 0, 320, 1)
 - U: (160, 120, 115200, 320, 1)
 - V: (160, 120, 76800, 320, 1)
- Więcej informacji:
<http://www.fourcc.org/yuv.php>



- Różne kombinacje przestrzeni barw z wzorcem organizacji w pamięci są identyfikowane przez tzw. kody 4CC
- 4CC = fourc character code, kod czteroliterowy
- 4-znakowy tekst (a równocześnie liczba 32-bitowa bez znaku w formacie big-endian)
- identyfikator wykorzystywany m.in. do określenia podformatu w plikach AVI
- najpopularniejsze formaty: AYUV, UYVU, YUY2, IYUV, I420, NV21

Rejestracja wideo: najpopularniejsze formaty RGB

12

- RGB24: 3B/piksel, każda składowa piksela wyrażona przez 1B; występuje w wariantach różniących się upakowaniem i kolejnością składowych
- ARGB: 4B/piksel, RGB24 + składowa przezroczystości (alpha)
- RGB15: 5b składowej koloru/piksel, upakowane w 2B (1b padding)
- RGB16: po 5b dla składowych R i B, 6b dla G, upakowane w 2B
- Więcej informacji:
<http://www.fourcc.org/rgb.php>

Rejestracja wideo: najpopularniejsze formaty YUV

13

- **I420** - format planarny z subsamplingiem 4:2:0, wykorzystywany w standardach JPEG, MJPEG, MPEG1, MPEG2 i MPEG4 (12bpp)
- **AYUV** - format YUV 4:4:4 z przeplotem i kanałem alpha, 4B makropiksele (32bpp)
- **UYVY** - format YUV 4:2:2 z przeplotem; 2B makropiksele (16bpp)
- Więcej informacji:
<http://www.fourcc.org/yuv.php>

- **Model Data-Push**

- Interfejs dźwiękowy lub wideo informuje aplikację o zgromadzeniu kolejnej „paczki” danych podczas rejestracji - przetwarzanie danych jest synchronizowane do zegara interfejsu
- Wymaga użycia mechanizmu callback lub innego sposobu notyfikacji asynchronicznych w trakcie rejestracji
- Aplikacja „zapisuje” kolejne paczki danych do interfejsu dźwiękowego w regularnych interwałach podczas odtwarzania - brak synchronizacji z zegarem interfejsu

- **Model Data-Pull**

- Aplikacja w określonych interwałach „odpytuje” interfejs dźwiękowy lub wideo o dostępność kolejnej porcji danych podczas rejestracji - brak synchronizacji z interfejsem
- Nie wymaga asynchronicznych notyfikacji w trakcie rejestracji
- Interfejs dźwiękowy „odpytuje” za pomocą mechanizmu notyfikacji asynchronicznych aplikację o kolejną porcję danych podczas odtwarzania - synchronizacja z zegarem interfejsu

Obsługa urządzeń multimedialnych: blokujące I/O

15

- Operacja odczytu lub zapisu urządzenia (np. interfejsu dźwiękowego) zatrzymuje wykonanie aplikacji do momentu, aż dostępna jest paczka danych (bufor audio, ramka wideo) lub zapisywana paczka została przekazana do urządzenia
- rejestracja: model pull; odtwarzanie: model push
- uniemożliwia to przetwarzanie danych w trakcie oczekiwania na kolejną porcję
- prowadzi do nieefektywnego wykorzystania wielowątkowości (typowo 1 wątek na 1 endpoint audio/wideo)

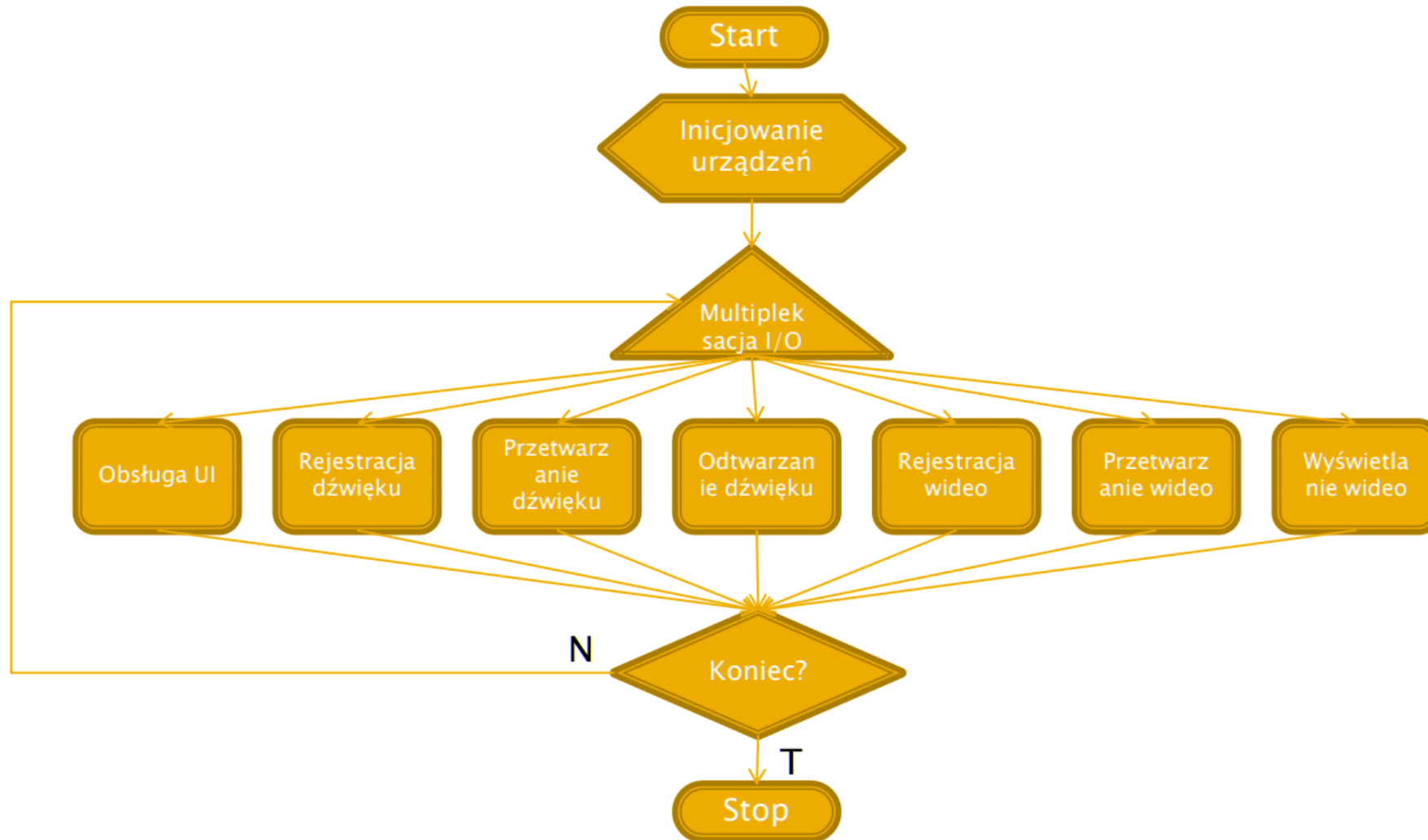
- Operacja zapisu lub odczytu powiedzie się tylko wtedy, gdy jest w danej chwili możliwa
- Występuje w połączeniu z mechanizmem multipleksacji I/O, zegarem lub mechanizmem notyfikacji o możliwości wykonania operacji
- Możliwe jest przetwarzanie danych i rejestracja/odtworzenie w ramach pojedynczego wątku

- **Dostarczają asynchronicznych powiadomień o zajściu zdarzenia związanego z interfejsem dźwiękowym lub wideo**
- **Wywołanie funkcji callback (z poziomu wątku urządzenia) - wymaga synchronizacji lub IPC**
- **Windows:**
 - Komunikaty okna (Windows messages) przetwarzane przez systemową pętlę komunikatów w aplikacjach GUI
 - Obiekty sygnalizowalne (synchronizacji) - zdarzenia, semafony itp.
- **Unix/Linux:**
 - Komunikaty systemu X/Windows
 - Sygnały

Obsługa urządzeń multimedialnych: multipleksacja I/O

18

- Usługa systemu operacyjnego, która pozwala na równoczesne blokowanie w oczekiwaniu na zakończenie kilku operacji, np. dostępność nowej paczki danych z urządzenia i upływanie zegara
- Pozwala uniknąć narzutu związanego z synchronizacją wielu wątków - zajście któregośkolwiek z oczekiwanych zdarzeń powoduje przerwanie blokowania i umożliwia jego obsługę
- Typowe realizacje: `select()`, `poll()`, `WaitForMultipleObjects()`



- **Dźwięk**

- Windows Core Audio API
 - DirectSound
 - DirectMusic
 - Windows multimedia waveXxx and mixerXxx functions
 - Media Foundation
- Windows SDK

- **Wideo**

- Video for Windows
- DirectShow

- **Dźwięk i wideo**

- Windows SDK
- Windows Multimedia API

- Zestaw funkcji języka C
- Obecny w każdym systemie Windows (od 3.0)
- Również Windows CE/Mobile
- Funkcjonalność
 - Rejestracja i odtwarzanie dźwięku próbkowanego
 - Obsługa miksera
 - Odtwarzanie MIDI
 - Obsługa skompresowanych formatów dźwięku poprzez systemowy ACM
 - Zapis i odczyt plików WAV
 - Obsługa zegara wysokiej precyzji

- Wykorzystywane mechanizmy notyfikacji
 - Callback
 - Komunikaty Windows
 - Obiekty synchronizacji
- Wysokie opóźnienia (kilkadziesiąt-kilkaset ms) wnoszone przez infrastrukturę dźwiękową systemu (ACM, mapper itp.)
- Rozwiązanie najbardziej uniwersalne i popularne
- Nie zalecane do zastosowań profesjonalnych

- Strumieniowe przesyłanie dźwięku bez opóźnień i błędów.
- Poprawiona niezawodność (wiele funkcji audio przeszło z trybu jądra do trybu użytkownika).
- Zwiększone bezpieczeństwo (przetwarzanie chronionej zawartości audio odbywa się w bezpiecznym procesie o niższych uprawnieniach).
- Przypisywanie poszczególnych ról systemowych (konsoli, multimediiów i komunikacji) poszczególnym urządzeniom audio.
- Abstrakcja oprogramowania dźwiękowych urządzeń końcowych (na przykład głośników, słuchawek i mikrofonów), którymi użytkownik manipuluje bezpośrednio.
- Zestaw bibliotek języka C++

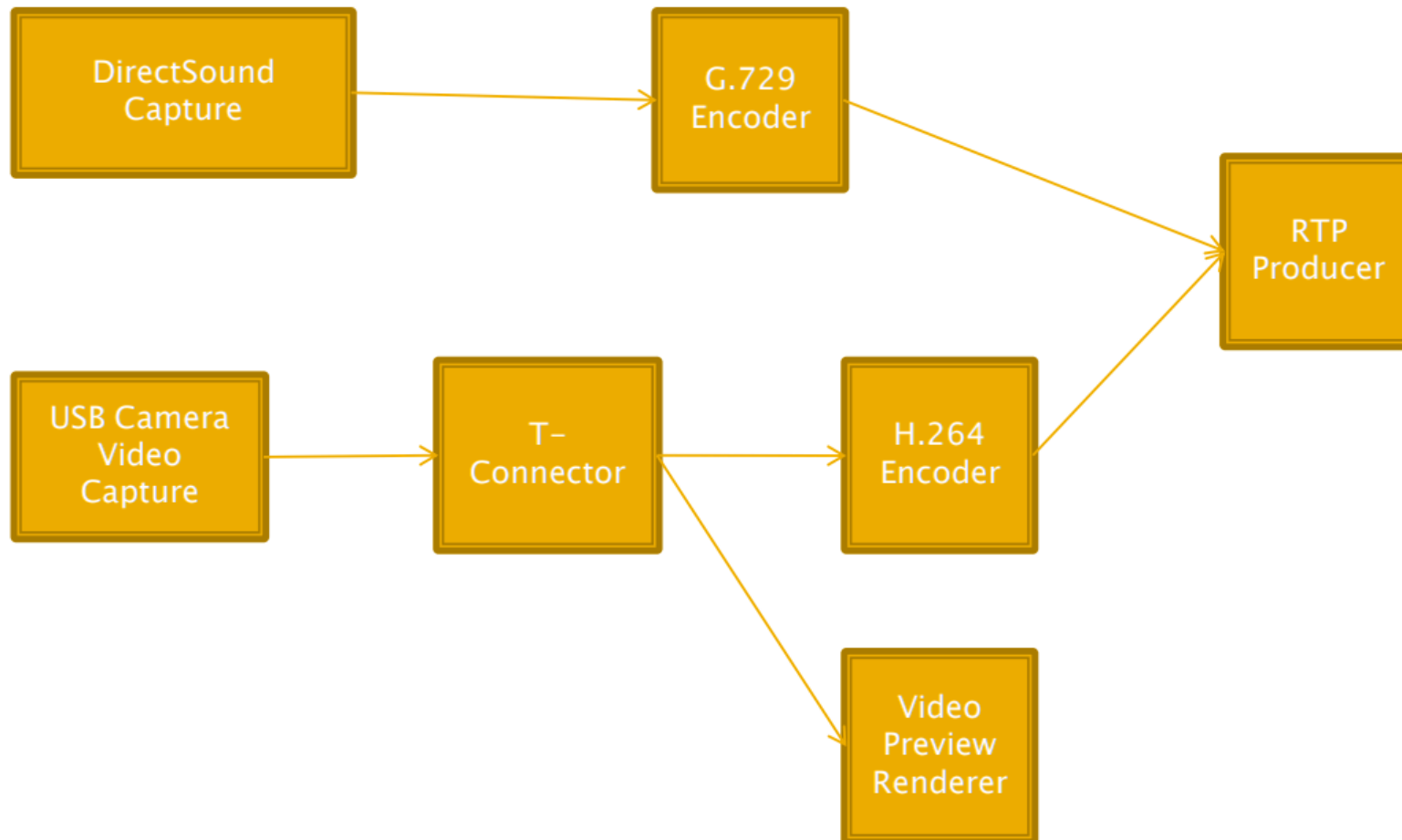
- Zestaw interfejsów modelu COM
- Obecny w systemach Windows od wersji 95
- Uproszczona wersja w systemach Windows CE/Mobile
- W starszych systemach wymaga instalacji dodatkowego oprogramowania (DirectX Runtime)
- Funkcjonalność
 - Rejestracja i odtwarzanie dźwięku próbkowanego
 - Pozycjonowanie dźwięku w przestrzeni 3D (DirectSound3D)
 - Współpraca z zaawansowanymi funkcjami i efektami sprzętowymi kart dźwiękowych (eg. hardware mixing)
 - Odtwarzanie MIDI (DirectMusic)

- Mechanizmy notyfikacji
 - Obiekty synchronizacji (Zdarzenia)
- Możliwość uzyskania niskich opóźnień (pojedyncze ms) - ze specjalnymi sterownikami i na niektórych systemach
- Począwszy od Windows XP - brak znaczących różnic wydajnościowych między WinMM API i DS.
- Interfejs projektowany pod kątem programowania gier komputerowych

- Zestaw funkcji języka C
- Odpowiednik WinMM API dla wideo
- Funkcjonalność
 - Przechwytywanie obrazu wideo (poprzez specjalne „okno wideo”)
 - Podgląd przechwytywanej sekwencji
- Ograniczona możliwość konfiguracji urządzeń - tylko ręcznie (przez użytkownika) poprzez specjalne okno konfiguracji

- Obsługa kompresji i dekompresji wideo (ramki w formatach kompresowanych)
- Mechanizmy notyfikacji
 - Callback
 - Komunikaty Windows
- API aktualnie jest deprecated (przestarzałe)

- System interfejsów COM umożliwiający tworzenie rozbudowanych aplikacji multimedialnych
- Opiera się na filtrach które są łączone w graf
- Każde urządzenie (interfejs audio, kamera, algorytm przetwarzania) stanowi filtr o określonej liczbie wejść i wyjść
- Przykładowe filtry wejścia
 - Kamera
 - Karta dźwiękowa
 - Odczyt pliku (AVI, MPEG, WAV itp.)
- Przykładowe filtry wyjścia
 - Video renderer (np. overlay)
 - Karta dźwiękowa
 - Zapis pliku



- System bardzo uniwersalny
- Kosztem wydajności - tworzenie i uruchomienie grafu jest szalenie nieefektywne
- Graf pochłania mnóstwo zasobów systemowych (pamięć, wątki)
- Technika „intelligent connect” - graf próbuje automatycznie wstawiać dodatkowe filtry między wejście i wyjście, aby dopasować akceptowane formaty danych

- Obecnie najbardziej popularne i zalecane
- Dedykowane systemowi Windows 10 ze wsparciem starszych (do wersji Server 2008)
- Głównie wykorzystywane do opracowywania aplikacji przy użyciu C++ lub platformy .NET
- Brak wsparcia dla odtwarzania DVD oraz brak Windows Media Center
- Łatwość instalacji i późniejszego użycia w Visual Studio
- Zdarzają się problemy w kompatybilności z wcześniejszymi wersjami Windowsa

- Advanced Linux Sound Architecture (ALSA)
- Video for Linux

- Zestaw funkcji języka C
- Składnik jądra systemu Linux od wersji 2.6.0, wcześniej opcjonalnie instalowalne moduły
- Funkcjonalność
 - Rejestracja i odtwarzanie dźwięku próbkowanego
 - Odtwarzanie MIDI
 - Obsługa kompresji dźwięku i efektów za pomocą specjalnych „pluginów”
 - Obsługa miksera

- Mechanizmy notyfikacji
 - Brak - możliwość multipleksacji I/O za pomocą funkcji poll()
- Możliwość uzyskania niskich opóźnień zależy od konkretnego sterownika i konfiguracji jądra Linuxa - specjalne dystrybucje do zastosowań profesjonalnych, które mają zmodyfikowany mechanizm przydziału czasu procesora

- **Moduły jądra systemu Linux**
 - V4l - kernel 2.1.x
 - V4l2 - kernel 2.5.x
- **Funkcjonalność**
 - Obsługa kart telewizyjnych
 - Przechwytywanie teletekstu
 - Obsługa radia
 - Przechwytywanie obrazu wideo

- **Rejestracja i odtwarzanie dźwięku**
 - **ASIO2** - sterowniki i biblioteka do ich obsługi firmy Steinberg, umożliwiają rejestrację i odtwarzanie dźwięku z bardzo małymi opóźnieniami na Windows i Apple - do zastosowań profesjonalnych
 - **PortAudio** - uniwersalna biblioteka rejestracji i odtwarzania dźwięku, wykorzystująca systemowe API. Działa na wielu systemach (m.in. Windows, Linux, BSD, Apple itp.)

- **Narzędzia przetwarzania dźwięku**
 - **libsndfile** - obsługa odczytu i zapisu wielu formatów plików dźwiękowych (biblioteka języka C)
 - **ffmpeg** - odczyt i zapis wielu formatów plików dźwiękowych i wideo, rejestracja wideo, kompresja audio i wideo
 - **libresample, libsamplerate** - konwersja częstotliwości próbkowania dźwięku
 - **fftw** - analiza widmowa (FFT)

- **Java Sound API (javax.sound.*)**
 - Rejestracja i odtwarzanie dźwięku próbkowanego
 - Odtwarzanie MIDI
 - Obsługa miksera
 - Duże opóźnienia wynikające z buforowania
 - Nieefektywna implementacja - nadmiarowa wielowątkowość
- **Java Media Framework (JMF)**
 - Rejestracja, odtwarzanie wideo
 - Kompresja audio i wideo
 - Streaming mediów (RTP)
 - Dla niektórych platform (Windows, Solaris) dostępne „natywne” implementacje - obsługa dodatkowych formatów, których przetwarzanie w Javie jest niemożliwe ze względów wydajnościowych

DZIĘKUJĘ ZA UWAGĘ