

Metody připojování periférií

BI-MPP

Přednáška 12

Ing. Miroslav Skrbek, Ph.D.

Katedra číslicového návrhu

Fakulta informačních technologií

České vysoké učení technické v Praze

Miroslav Skrbek ©2010-2021

ZS2021/22



Evropský sociální fond
Praha & EU: Investujeme do vaší budoucnosti

Agenda

- Video for Linux
- Windows Multimedia API
- XAudio2

Literatura

- Gook, M.: Hardwarová rozhraní – Průvodce programátora. Computer Press, Brno 2006. ISBN 80-251-1019-2
- LINUX MEDIA INFRASTRUCTURE API. LinuxTV Developers 2009-2011.
(<http://linuxtv.org/downloads/v4l-dvb-apis/>)
- Windows Multimedia Functions
(<http://msdn.microsoft.com/en-us/library/ms925318.aspx>)

Video for Linux (v4l2)

- Linuxový standard pro přístup k video zařízením
- Zdroj: *Schimek, M., H.: Video for Linux Two API Specification, Revision 0.24.*
- Rozhraní definováno nad souborovými operacemi open, read, write, ioctl, close.
- Pro rychlejší výměnu dat je možno přímé mapování bufferů do paměťového prostoru aplikace (mmap).
- Pro řízení video zařízení je definována sada funkcí řízených přes ioctl

Video zařízení v Linuxu

```
/dev/video      # defaultní zařízení  
                # typicky link na  
                # /dev/video0
```

jednotlivá video zařízení

/dev/video0

/dev/video1

/dev/video2

/dev/video3

...

Otevření video zařízení

```
int fd = open("/dev/video0");  
if (fd < 0) {  
    // signalizace chyby  
}
```

Proměnná fd (file descriptor) je dále užívána jako odkaz na otevřené zařízení

Zjištění informací o video zařízení

```
struct v4l2_capability cap;  
  
If (ioctl(fd, VIDIOC_QUERYCAP, &cap) < 0 {  
    // signalizace chyby  
}
```

Cap obsahuje položku capabilities s hodnotami:

- V4L2_CAP_VIDEO_CAPTURE - má rozhraní pro snímání obrazu (capture)
- V4L2_CAP_VIDEO_OVERLAY – má rozhraní přímé ukládání obrazu do video paměti grafické karty
- V4L2_CAP_AUDIO – zařízení má audio vstup/výstup

Zjištění počtu a nastavení video vstupu

VIDIOC_ENUMINPUT zjišťuje počet video vstupů. Argumentem je `struct v4l2_input`

VIDIOC_S_INPUT nastavuje požadovaný vstup. Argumentem je `int*`

Metody čtení/zápisu dat z/do video zařízení

- Read/write
 - Metoda užívá standardních read/write operací. Může mít přidanou režii pro kopírování dat mezi user space a kernel space, pokud driver nepoužívá DMA
- Streaming (paměťově mapované buffery)
 - Používá funkce mmap pro mapování fyzické paměti do virtuálního adresního prostoru
 - Video buffery jsou alokovány ve fyzické paměti
- Streaming (uživatelské ukazatele)
 - Video buffery jsou alokovány v aplikaci (user space)
 - Do driveru se předávají pouze ukazatele na buffery

Streaming

(paměťově mapované buffery)

- ioctl VIDIOC_REQBUFS se použije pro alokaci video bufferů ve fyzické paměti na straně ovladače. Argumentem je požadovaný počet bufferů. Počet alokovaných bufferů může být menší než požadovaný
- Vytvoří se pole popisovačů bufferů. Popisovač obsahuje adresu počátku bufferu v user space a jeho délku.
- Pole se vyplní opakovaným voláním ioctl VIDIOC_QUERYBUF, které dodá délku bufferu. Adresu počátku bufferu zjistíme funkcí mmap, která buffer namapuje do aplikace. Adresu doplníme do popisovače
- Aplikace volá VIDIOC_QBUF ioctl (enqueue the buffer), které zařadí buffer do fronty v ovladači a ten ji pak buffer vyplní snímkem. Vyplněný buffer může aplikace převzít voláním VIDIOC_DQBUF ioctl (dequeue the buffer).
- Typicky máme-li 5 bufferů, pak pro všechny voláme VIDIOC_QBUF ioctl, dále zavoláme VIDIOC_STREAMON a pak následuje čtecí smyčka, kde testujeme stav bufferu (VIDIOC_QUERYBUF ioctl) a přejímáme voláním VIDIOC_DQBUF ioctl.

Windows Multimedia API

- API pro přehrávání a snímání audia a videa
- Dnes se nahrazuje jednoduššími a multiplatformními (PC a XBox) rozhraními z DirectX (XAudio2)
- Přehrávání videa i audia

Otevření streamu pro záznam

```
result = waveInOpen(&hWaveIn, deviceId, &pFormat,  
                    (DWORD_PTR) waveInProc, (DWORD)  
                    & audio_buffers,  
                    WAVE_FORMAT_DIRECT  
                    | WAVE_MAPPED |  
                    CALLBACK_FUNCTION);  
if (result != MMSYSERR_NOERROR) {  
    return 0;  
}
```

Příprava hlavičky bufferu

```
waveInPrepareHeader(hWaveIn, hdr, sizeof (WAVEHDR));
```

Tato funkce připravuje hlavičku bufferu pro to, aby mohl být použit pro ukládání nebo přehrávání audio dat. Tato funkce koncentruje všechny časově náročné inicializace jak na straně audio driveru, tak operačního systému.

Zařazení bufferu do fronty

```
// přidání bufferu do fronty  
waveInAddBuffer(hWaveIn, &audio_buffers.WaveInHdr[i],  
sizeof (WAVEHDR));
```

Po naplnění bufferu (indikováno např. voláním callback funkce) aplikace naplní/překopíruje audio data a může znovu ten samý buffer zařadit do fronty funkcí `waveInAddBuffer`. Toto se děje opakovaně nad množinou třeba 5 bufferů.

Start a stop záznamu

```
// start záznamu
```

```
waveInStart (hWaveIn) ;
```

```
// stop záznamu
```

```
waveInReset (hWaveIn) ;
```

Funkce pro přehrávání

- `waveOutOpen`
 - Otevření výstupního streamu
- `waveOutPrepareHeader`
 - Inicializace bufferu
- `waveOutWrite`
 - Zařazení bufferu do fronty pro přehrávání

Další funkce lze nalézt v MSDN

<http://msdn.microsoft.com/en-us/library/ms925318.aspx>

XAudio2

- Audio API – součást DirectX
- Podporuje míchání hlasů
 - SourceVoice (zdrojové hlasy)
 - Submix Voice (pomocné míchání uvnitř struktury míchaných hlasů)
 - Master Voice (výsledné míchání a výstup)
- Nepodporuje audio input