

Processamento de áudio em tempo real em sistemas Android

André Jucovsky Bianchi
ajb@ime.usp.br

Departamento de Ciência da Computação
Instituto de Matemática e Estatística
Universidade de São Paulo

4 de maio de 2012

Estrutura da apresentação

Introdução

Fontes de sinais de áudio

Agendamento

Manipulação e reprodução do sinal

Conclusões

Introdução

Dispositivos com o sistema Android:

- ▶ Computação.
- ▶ Comunicação via ondas de rádio.
- ▶ Sensores.
- ▶ Abstração no sistema operacional.

Nosso foco na programação para Android:

- ▶ Sistema padrão.
- ▶ Java (incluindo a possibilidade de código nativo).

Outras iniciativas:

- ▶ Pure Data (pdlib) [Brinkmann 2012].
- ▶ Csound [YI e LAZZARINI 2012].

Fontes de sinais de áudio

Possíveis fontes de sinais de áudio são:

- ▶ Microfone.
- ▶ Arquivos armazenados na memória.
- ▶ Transmissão de dados via rádio.

Fontes de sinais de áudio

Microfone

Formas de acesso ao sinal do microfone:

- ▶ MIC.
- ▶ CAMCORDER.
- ▶ VOICE_DOWNLINK.
- ▶ VOICE_UPLINK.
- ▶ VOICE_CALL.
- ▶ VOICE_COMMUNICATION.
- ▶ VOICE_RECOGNITION.

Fontes de sinais de áudio

Microfone: `AudioRecord`

```
1 AudioRecord recorder = new AudioRecord(  
2   AudioSource.MIC, // Fonte de audio.  
3   44100,           // Taxa de amostragem (Hz).  
4   AudioFormat.CHANNEL_IN_MONO, // Numero de canais.  
5   AudioFormat.ENCODING_PCM_16BIT, // Codificacao.  
6   AudioRecord.getMinBufferSize( // Tamanho do buffer.  
7     44100,  
8     AudioFormat.CHANNEL_IN_MONO,  
9     AudioFormat.ENCODING_PCM_16BIT));
```

Observações:

- ▶ As configurações possíveis dependem do modelo do aparelho.
- ▶ O buffer interno pode ajudar a evitar perda de amostras.
- ▶ Para fazer a leitura das amostras: `AudioRecord.read()`.

Fontes de sinais de áudio

Microfone: laço de leitura

```
1 public void readLoop(short[] inputBuffer,
2                       int readBlockSize) {
3     while (isRunning) {
4         recorder.read(
5             inputBuffer, // buffer para as amostras.
6             ((j++) * readBlockSize) % buffer.length, // idx
7             readBlockSize); // tamanho do bloco.
8     }
9 }
```

Observações:

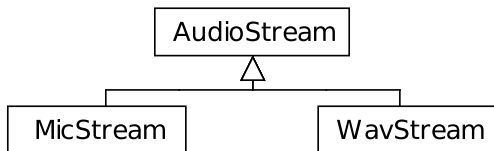
- ▶ `AudioRecord.read()` é bloqueante.
- ▶ Processamento de amostras poderia ser incluído neste laço.

Fontes de sinais de áudio

Arquivos

Observações sobre sinais armazenados em arquivos:

- ▶ Não há suporte para leitura (de amostras) de arquivos na API do Android.
- ▶ Uma possibilidade de implementação é a seguinte:



Agendamento

Características do sinal:

- ▶ Taxa de amostragem (R).
- ▶ Número de canais.
- ▶ Codificação das amostras.

Características do processamento:

- ▶ Tamanho do bloco de processamento (N).
- ▶ Fator de sobreposição (M).

Fatos:

- ▶ Atraso mínimo de $\frac{N}{R}$ segundos.
- ▶ Período do ciclo de processamento: $\frac{N}{MR}$ segundos.
- ▶ Intervalo de amostras processadas no j -ésimo ciclo:
 $[j\frac{N}{M}, j\frac{N}{M} + N - 1]$.
- ▶ Instante de execução do j -ésimo ciclo: $t_j = \frac{N}{R} + j\frac{N}{MR}$.

Agendamento

```
1 public void scheduleDspCallback() {
2     System.gc(); // recolhimento de lixo
3     try {
4         scheduler = Executors.newScheduledThreadPool(1);
5         dspTask = scheduler.scheduleAtFixedRate(
6             dspCallback, // a funcao de processamento.
7             (float) N/R*Math.pow(10,9), // atraso.
8             (float) N/(MR)*Math.pow(10,9), // intervalo.
9             TimeUnit.NANOSECONDS); // unidade de tempo.
10    } catch (Exception e) {
11        e.printStackTrace();
12    }
13 }
```

Observações:

- ▶ dspCallback é um objeto do tipo Runnable.
- ▶ scheduler é um objeto do tipo ScheduledExecutorService.
- ▶ *“This class does not offer guarantees about the real-time nature of task scheduling”*.

Manipulação e reprodução do sinal

```
1 AudioTrack track = new AudioTrack(  
2     AudioManager.STREAM_MUSIC,      // stream de audio.  
3     44100,                          // taxa de amostragem.  
4     AudioFormat.CHANNEL_OUT_MONO,   // numero de canais.  
5     AudioFormat.ENCODING_PCM_16BIT, // codificacao.  
6     audioStream.getMinBufferSize(), // tamanho do buffer.  
7     AudioTrack.MODE_STREAMING);     // modo de operacao
```

Observações:

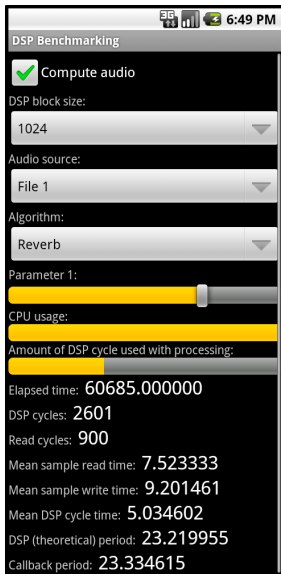
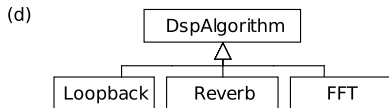
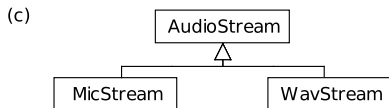
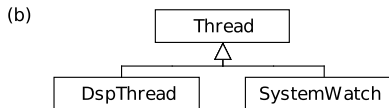
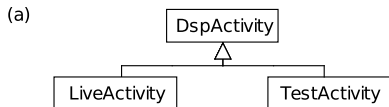
- ▶ `audioStream.getMinBufferSize()` encapsula uma chamada com as mesmas configurações de sinal (taxa de amostragem, número de canais e codificação).
- ▶ `AudioTrack` possui diferentes modos de operação: *static* e *streaming*.

Manipulação e reprodução do sinal

```
1 short inputBuffer; // buffer circular de entrada.
2 short outputBuffer; // buffer circular de saída.
3 int j; // indice de leitura dos buffers.
4 int L = inputBuffer.length;
5 Runnable dspCallback = new Runnable() {
6     public void run() {
7         // executa o algoritmo escolhido
8         dspAlgorithm.perform(inputBuffer, j);
9         // overlap-add
10        for (int i = 0; i < N; i++) {
11            if (i >= N-N/M)
12                outputBuffer[(j*N/M + i) % L] = 0;
13            outputBuffer[(j*N/M+i) % L] +=
14                (inputBuffer[(j*N/M+i) % L] * Short.MAX_VALUE);
15        }
16        // escreve as amostras para reproducao
17        track.write(outputBuffer, ((j++)*N/M) % L, N/M);
18    }
19 };
```

Considerações finais

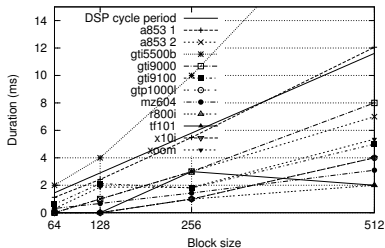
Arquitetura para testes



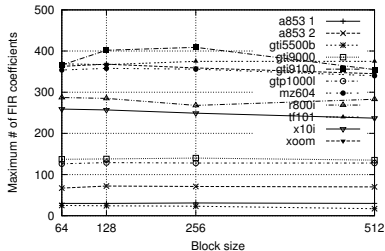
Considerações finais

Alguns resultados

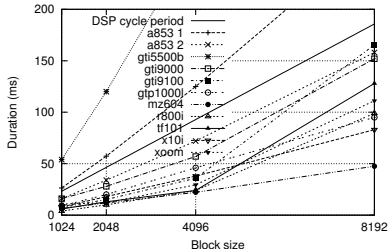
Callback times for fft on each device (1/2)



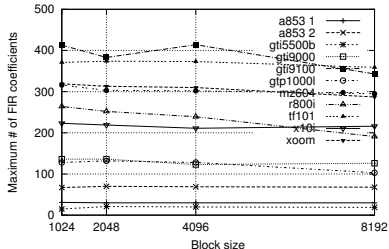
Maximum number of FIR coefficients per device per block (1/2)



Callback times for fft on each device (2/2)



Maximum number of FIR coefficients per device per block (2/2)



Considerações finais



Outras abordagens

Outras abordagens possíveis para processamento em tempo real em plataformas Android:

- ▶ Classe `AudioEffect`.
- ▶ Código nativo (JNI).

Obrigado pela atenção.

Bibliografia:

-  BRINKMANN, P. *Making Musical Apps*. [S.l.]: O'Reilly Media, 2012.
-  YI, S.; LAZZARINI, V. Csound for android. *Proceedings of the Linux Audio Conference 2012*, 2012.

Dados de contato:

- ▶ Meu email: ajb@ime.usp.br
- ▶ Esta apresentação: <http://www.ime.usp.br/~ajb/>
- ▶ CM no IME: <http://compmus.ime.usp.br/>