

Conversão e processamento de arquivos de vídeo com o transcode

Tudo se transforma

O transcode é o canivete suíço da manipulação de arquivos multimídia no Linux. Com sua arquitetura modular, ele oferece ao usuário muito mais do que a simples conversão entre formatos...

POR JÖRN REDER

O *transcode* [1] é um verdadeiro canivete suíço que converte arquivos de um formato de vídeo para outro. Também pode ser usado para outros tipos de manipulação de vídeo, como redimensionamento de quadros. Se você está esperando que uma ferramenta de conversão de vídeos tenha uma interface gráfica bonitinha, prepare-se para se desapontar. O programa funciona apenas na linha de comando e é operado com uma porção de parâmetros em modo texto. Como a maioria das ferramentas que não possuem interface gráfica, a maior vantagem do *transcode* é sua flexibilidade e seu maior ponto fraco é a usabilidade. Os que preferem fazer tudo graficamente podem escolher numa lista um *front-end* que ofereça uma interface amigável para seus recursos especiais

(ver **tabela 1**). Mas nenhum desses programas consegue abarcar plenamente o poder dessa venerável ferramenta.

O *transcode* tem uma estrutura estritamente modular. As funções críticas residem em módulos externos que só são chamados quando necessário. Há três tipos diferentes de módulo, que correspondem às três etapas do processo de conversão de arquivos quando feito por esse programa:

- *Input/Decoding* (Entrada/Decodificação)
- *Modification/Filtering* (Modificação/Filtragem)
- *Output/Encoding* (Saída/Codificação)

Os módulos de entrada (ver **tabela 2**) têm a função de decodificar arquivos de vídeo e áudio e entregá-los ao *transcode* “crus” sem qualquer pós-processamento ou formatação (*raw format*). Há diferentes

módulos para *codecs* individuais de áudio e vídeo, assim como para captura de vídeo vindo das populares placas de TV.

A etapa seguinte aplica módulos de filtro (ver **tabela 3**) aos dados brutos entregues pelos módulos de entrada. Os filtros não apenas modificam os dados de imagem e som existentes, como também podem adicionar ou remover quadros para alterar a sincronia entre o vídeo e o áudio.

Na etapa final, o *transcode* passa os dados para um ou mais módulos de saída (ver **tabela 4**), que os codificam por meio de uma variedade de *codecs* de áudio e vídeo e criam o arquivo no formato desejado.

Conversões simples

O seguinte comando do *transcode* converte um arquivo MPEG1 ou MPEG2 para um arquivo AVI no formato MPEG4: →

Tabela 1: Front-ends para o transcode

Nome	Site oficial	Descrição
dvd::rip	http://www.exit1.org/dvdrip/	Interface gráfica especializada na extração (<i>ripping</i>) de DVDs e conversão para outros formatos (como DivX).
g4l	http://gv4l.sourceforge.net/	Captura e manipula vídeo de dispositivos <i>Video4Linux</i> , como <i>tuners</i> e <i>webcams</i> . Pode ser usado para capturar e manipular programas de TV.
ripmake	http://www.lallafa.de/bp/ripmake.html	Programa em linha de comando que facilita tarefas comuns do <i>transcode</i> .
kavi2svcd	http://www.cornelinux.de/web/linux/kavi2svcd/index.html	Converte arquivos AVI em MPEGs para a criação de discos SVCD.

```
# transcode -V -i video_original 2
-o video_resultante.avi -y xvid4 2
-w 500 -b 48
```

O *transcode* usa a biblioteca *XviD* nativa do Linux para esse trabalho. A biblioteca é veloz e confiável, e quando usada sabiamente pode superar em muito os *codecs* comerciais no tocante à qualidade. Este exemplo simples demonstra as duas opções mais importantes: *-i* e *-o* indicam os arquivos de entrada e saída, respectivamente; *-y* ajusta o filtro de exportação de vídeo para *xvid4* e *-w* configura a taxa de bits (ou *bitrate*) do vídeo para 500 kbit/s. Como o *transcode* exporta áudio em MP3 a 128 kbit/s por padrão, precisamos apenas ajustar a taxa de bits da faixa de som para 48 kbit/s usando a opção *-b*. Finalmente, a opção *-V* diz ao *transcode* que use o espaço de cor YUV e não RGB. Isso acelera os cálculos internos em quase 30%. Porém, há alguns filtros que só funcionam no espaço de cor RGB. Se esse for o caso, o *transcode* emitirá uma mensagem de erro se você tentar estipular a opção *-V*.

Adivinhando a taxa de bits

A taxa de bits do vídeo é o fator que decide a qualidade do produto final. Você precisa de diversos parâmetros, como a altura e a largura (em pixels) do quadro e o número de quadros por segundo para calcular o valor BPP (Bits Por Pixel). A fórmula é: *Taxa de Bits x 1000 / (Altura x Largura x Quadros por segundo)*.

Obviamente, a qualidade de um vídeo depende em grande parte da qualidade do material original (cenas rápidas/lentas, muito/pouco contraste etc.); mas você pode adotar como regra que um valor de BPP de cerca de 0,20 oferecerá uma qualidade aceitável e evitará a artificialidade. Valores abaixo de 0,15 afetarão a qualidade da imagem visivelmente. O *transcode*, na verdade, ajusta o valor de BPP segundo as configurações que você selecionar (na linha *V: bits/pixel*). Se o valor for muito baixo, é melhor sair do *transcode* pressionando *[Ctrl]+[C]* e tentar novamente com um valor mais alto.

Melhor prevenir do que remediar

O *transcode* suporta a conversão em múltiplos passos para *codecs* MPEG4. Para conseguir uma melhor distribuição da taxa de bits ao longo de um vídeo, o pro-

grama primeiro analisa o material e cria um arquivo de *log*. O segundo passo analisa o arquivo de *log* e faz o trabalho de conversão. Você pode habilitar os passos

múltiplos especificando a opção *-R*; é preciso especificar se esse é o primeiro ou o segundo passo. A **listagem 1** traz um exemplo.

Tabela 2: Módulos de importação do *transcode*

Módulo	Descrição
:dvd	Lê diretamente de um DVD de vídeo. A opção <i>-i</i> precisa apontar para um nome de arquivo de dispositivo ou para o nome do diretório com o sistema de arquivos do DVD.
vob	Lê um arquivo VOB (<i>Video Object</i>) extraído de um DVD. Você pode criar o arquivo usando o comando <i>tcclat</i> do <i>transcode</i> .
dv	Lê vídeo no formato DV.
Ffmpeg	Lê dados em um dos formatos suportados pela biblioteca <i>ffmpeg</i> . O formato será atribuído automaticamente.
mplayer	Chama o <i>Mplayer</i> para decodificar o vídeo. O <i>transcode</i> pode, portanto, lidar com qualquer codec suportado pelo <Mplayer>.
v41 / v412	Lê um sinal de vídeo diretamente de um dispositivo <i>Video4Linux</i> , como uma webcam ou placa de TV.
xvid	Lê vídeo no formato MPEG4.
mpeg2	importação de vídeo MPEG2.

Tabela 3: Filtros do *transcode*

Filtro	Descrição
32detect	Deteção de quadros entrelaçados. É preciso especificar a opção <i>verbose=1</i> para descobrir se é necessário desentrelaçar a imagem.
smartdeinter	Desentrelaçador de imagens RGB. Derivado do <i>smart deinterlacer</i> do <i>VirtualDub</i>
smartyuv	Desentrelaçador de imagens YUV. Necessita da opção <i>-V</i> .
modfps	Conversor de taxa de quadros, também chamada de <i>framerate</i> (por exemplo, dos 24 quadros por segundo de um filme para os 29,97 quadros usados na TV nos padrões PAL-M ou NTSC). É preciso especificar a taxa de quadros desejada com a opção <i>--export_fps</i> .
logo	Aplica um logotipo à imagem, pode ser usado como marca d'água.
logoaway	Remove o logotipo da estação de TV no cantinho da imagem.
normalize	Normaliza o volume.
yuvdenoise	Filtro de ruídos em imagens YUV. Necessita da opção <i>-V</i> .
detectclipping	Detecta, com <i>-j</i> , ou remove, com <i>-Y</i> , as barras negras no topo e na parte de baixo da tela.
pv	Abre uma janela de visualização (<i>preview</i>). O servidor X deve suportar a extensão XV.

Tabela 4: Módulos de exportação do *transcode*

Módulo	Descrição
xvid / xvid4	O codec <i>XviD</i> , derivado Open Source do <i>DivX</i> escreve arquivos no formato MPEG4. O <i>transcode</i> trabalha com diversas versões dessa biblioteca. O <i>transcode</i> 0.6.12 usa como padrão o <i>XviD</i> 0.9. Recomenda-se usar sempre a versão mais atual do <i>XviD</i> (no momento, a 1.1.0-beta 1) com o módulo <i>xvid4</i> .
divx5	Usa o codec comercial <i>DivX</i> , fornecido pela <i>DivX Networks</i> (www.divx.com). Embora esse codec esteja disponível em formato nativo para o Linux, seu uso não é recomendado devido a algumas falhas e ao fato de seu código-fonte ser fechado.
ffmpeg	Quaisquer codecs reconhecidos pela biblioteca <i>ffmpeg</i> . Para obter uma lista dos codecs disponíveis, digite <i>transcode -y ffmpeg -F list</i> . Selecione os codecs desejados especificando a opção <i>-F</i> .
mpeg2enc, mp2enc	MPEG2 vídeo, MP2 áudio. O utilitário <i>tcmlp</i> funde os arquivos <i>.m2v</i> e <i>.m2a</i> resultantes para criar um arquivo com a extensão <i>.mpeg</i> , pronto para ser usado em um VCD, SVCD ou DVD.
wav	módulo de exportação somente de áudio, gera arquivos no formato WAV.

Listagem 1: Exemplo de codificação em múltiplos passos

```
# transcode -V -i video_original.mpeg -o video_resultante.avi -y xvid4,null -w 500 -b 128 -R 1
# transcode -V -i video_original.mpeg -o video_resultante.avi -y xvid4 -w 500 -b 128 -R 2
```

Uma vez que o primeiro passo só fará a análise do vídeo, a opção `-y xvid4,null` desabilita a função de exportação de áudio. Isso acelera o processo em mais de dez por cento.

Análise versus tentativa e erro

Como a conversão adequada de vídeos depende de material-fonte de alta qualidade, o `transcode` oferece uma coleção de ferramentas para análise de arquivos de vídeo e exibição de suas características técnicas críticas. O comando `tcprobe` dá uma rápida visão geral (veja o exemplo no quadro “Resultados do `tcprobe`”).

De acordo com o `tcprobe`, o arquivo fonte tem uma resolução de 320x240 pixels e uma taxa de quadros de 29.97 (comum a sistemas NTSC) quadros por segundo. Além disso, o arquivo tem uma faixa de áudio mono de 44.1 khz com uma taxa de amostragem de 16-bit (como indicado por `-e 44100,16,1` – um arquivo estéreo mostraria um 2 como último dígito). A faixa de áudio é codificada em MP2 – infelizmente isso não é óbvio; é preciso analisar o parâmetro `-n` (nesse caso `-n 0x50`) para descobri-lo. `0x55` significa codificação MP3 e `0x2000` significa áudio em AC3. A taxa de bits do áudio está configurada para 48 kbit/s. Como o `transcode` usa a codificação de 128 kbit/s por padrão, é boa política diminuir manualmente essa taxa para 48 kbit/s, uma vez que um valor mais alto não aumentará a qualidade de saída, mas certamente aumenta o tamanho do arquivo.

Os valores entre colchetes nas informações mostradas pelo `tcprobe` indica as configurações padrão do `transcode`. O programa usará esses valores se você lhe passar dados brutos com valores de formatação que ele não reconheça.

Pequeno, mas poderoso

A capacidade de redimensionar imagens é outro recurso muito útil – ele permite criar um preview de um arquivo muito grande disponível para download. O `transcode` oferece para isso diversas funções; a mais simples usa a opção `-Z` e pede os valores de largura e altura. O comando a seguir

reduz o tamanho da imagem à metade, para 160x120 pixels, sem usar o método de múltiplos passos:

```
# transcode -V -i video_original.mpeg -o resultado_pequeno.avi -y xvid4 -w 125 -b 48 -Z 160x120,fast
```

O parâmetro `fast` manda o `transcode` usar um algoritmo interno de redimensionamento que pode afetar levemente a qualidade da imagem e que tem algumas restrições. Por exemplo, a largura e a altura da imagem têm de ser divisíveis por 8. Se não forem, o programa voltará ao método padrão. A opção `-w 125` também é importante; ela reduz a taxa de bits do vídeo. Como o exemplo reduz a a largura e a altura, a imagem só ocupará um quarto do tamanho original; isso equivale a um quarto da taxa de bits.

Advertência: Gravação da TV

Usuários com gravadores de vídeo digitais – as chamados DVRs, ou Digital Video Recorders – rodando no Linux podem querer armazenar as gravações feitas a partir da TV em um formato de alta compressão como XviD ou MPEG4. O `transcode` também pode ajudar nisso; porém, precisa de uma ajudazinha do `VDRsync` [2]. Embora o `transcode` consiga lidar com um arquivo DVR diretamente, problemas

com sincronização de áudio e vídeo são bastante comuns; o `Vdrsync` pode resolver esses problemas antes mesmo de você começar. O seguinte comando cria dois arquivos a partir de uma gravação DVR para permitir o armazenamento separado das faixas de áudio e de vídeo:

```
# vdrsync.pl /vdrdata/Dragonheart/2004-08-08.20\13.50.50.rec/
```

O nome do arquivo de saída pode variar. Neste exemplo, o script criou um arquivo de vídeo chamado `e4.mpv` e um arquivo de áudio chamado `c0.mpa`. Agora o `transcode` pode manipular a ambos.

Visão limitada

Se o material do filme estiver em formato *Letterbox*, você vai notar barras pretas no alto e embaixo da tela. Como a barra preta é um desperdício de espaço, provavelmente você vai preferir que o `transcode` a elimine. Esse é um processo em duas etapas: primeiramente, o `transcode` precisa analisar o tamanho das barras e descobrir suas posições exatas. Para fazê-lo, ele usa o filtro `detectclipping`. A [listagem 2](#) traz um exemplo.

O que o `transcode` faz aqui, na verdade, é inspecionar cada quadro individualmente e mostrar os resultados na linha de comando. Quando o resultado parar de mudar, o `transcode` terá encontrado as

Quadro 1: Resultados `tcprobe`

```
# tcprobe -i video_original.mpeg
[tcprobe] MPEG program stream (PS)
[tcprobe] summary for video_original.mpeg, (*) = not default, 0 = not detected
import frame size: -g 320x240 [720x576] (*)
aspect ratio: 1:1
frame rate: -f 29.970 [25.000] frc=4 (*)
audio track: -a 0 [0] -e 44100,16,1 [48000,16,2] -n 0x50 [0x2000] (*)
PTS=370.6876, bitrate=48 kbit/s
```

Listagem 2: Analisando as barras pretas

```
# transcode -V -i e4.mpv -J detectclipping
...
[detectclipping#0] valid area: X: 9..712 Y: 57..519 -> -j 58,10,56,6
...
```

configurações corretas. Em nosso exemplo, o parâmetro `-j 58,10,56,6` manda o programa remover 58 linhas do topo, 10 à esquerda, as últimas 56 e as 6 linhas à direita da imagem.

Infelizmente, você não pode usar esses valores assim como estão, pois o tamanho final da imagem não seria divisível por 16 (exigência da maioria dos codificadores, e o *XviD* não é exceção). Algumas contas simples ajudarão. Para ficar mais simples ainda, vamos apenas ignorar os poucos pixels à direita e à esquerda e nos concentrar em cima e embaixo. Se removermos 56 linhas em vez de 58, estaremos removendo um total de 112 linhas. 576 linhas num quadro PAL, menos 112, dá 464, que é divisível por 16: dá certo. Resta-nos portanto `-j 56,0,56,0`; podemos encurtar isso para: `-j 56`. O programa, por padrão, corta a imagem simetricamente.

Entrelaçamento

Entrelaçamento é uma palavra comum em tecnologia de vídeo. Quando o vídeo está entrelaçado, as linhas pares e ímpares da imagem são exibidas alternadamente: em um quadro só as pares, no outro só as ímpares – o que é bem normal nas TVs. Quando você exibe uma imagem num monitor de PC, os resultados são ruins porque os objetos em movimento tendem a criar “fantasmas” ou o chamado “efeito pente”. Para rebater esse problema, a maioria dos reprodutores ou conversores de vídeo usa uma função de desentrelaçamento. O *transcode* usa o filtro *32detect* (ver [listagem 3](#)) para detectar se o vídeo inclui quadros entrelaçados.

O *transcode* mostra uma linha para cada quadro. Se houver quadros com `interlaced = yes`, é bom usar o filtro de desentrelaçamento. O lado ruim é que a codificação leva muito mais tempo neste caso. O *transcode* tem um filtro *smartyuv*, a versão com velocidade otimizada do filtro *smart deinterlacer* do *VirtualDub* para o espaço de cor YUV. O *VirtualDub* é um popular programa do Windows® que faz algo similar ao *transcode*. O filtro tem uma

Quadro 2: Guia de instalação do *transcode*

Se a sua distribuição Linux tem pacotes do *transcode* prontos para rodar, você tem a obrigação moral de usá-los. Compilar o programa a partir do código fonte exige muita experiência, já que o programa usa funções de uma porção de bibliotecas. De acordo com o gerenciador de pacotes do Debian, o *transcode* tem 44 dependências. Este artigo baseia-se na versão 0.6.12, publicada em janeiro e que pode ser baixada de [1]. Há binários disponíveis para as seguintes distribuições:

⇒ Debian: <ftp://ftp.nerim.net/debian-marillat/index.html>

⇒ Gentoo: <http://packages.gentoo.org/>

⇒ RedHat: <http://freshrpms.net/>

⇒ Mandrake: <http://plf.zarb.org/>

⇒ SuSE: <http://packman.links2linux.org/>

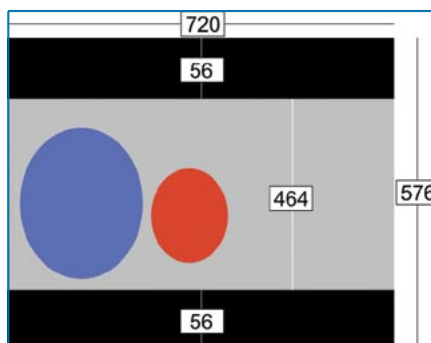


Figura 1: Um vídeo PAL com resolução máxima tem 720x576 pixels. A imagem em 4:3 fica distorcida porque 720x576 equivale a uma proporção de 5:4. Isso faz com que os personagens do filme pareçam ter a cabeça bem comprida se você for assisti-lo no formato original em um PC.

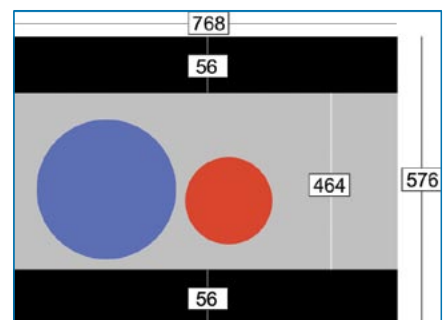


Figura 2: Para remover a distorção, a imagem precisa ser esticada lateralmente – isso torna as cabeças redondas novamente.

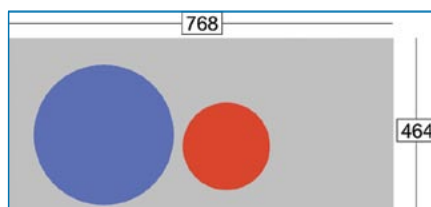


Figura 3: As barras pretas gastam bits valiosos; remova-as para economizar espaço.

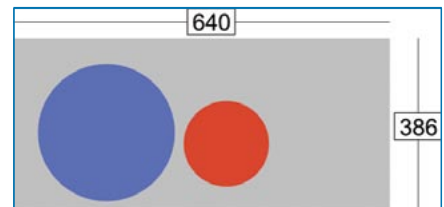


Figura 4: Para poupar ainda mais espaço, você pode reduzir a resolução. Nosso exemplo usa uma largura de 640 pixels, que não deixa de oferecer uma boa resolução.

porção de opções que você pode explorar, mas os ajustes padrão funcionam bem na maioria dos casos.

Mastigando números

Para poupar ainda mais espaço, queremos não apenas remover as barras pretas como também diminuir o tamanho

da imagem. Vídeo no padrão PAL tem uma resolução original de 720x576 pixels, mas o tamanho da imagem é distorcido de modo amorfo quando ela é exibida numa tela de PC, já que a proporção é de 5:4, e não de 4:3 (usada nos monitores). A maior parte dos *media players* corrigirá isso automaticamente, mas o objetivo é produzir um arquivo AVI com a proporção amigável ao PC (e às TVs analógicas que representam a esmagadora maioria dos aparelhos no Brasil) de 4:3. A distorção é ainda maior com vídeo no formato 16:9. Claro que se podem aplicar os cálculos para 16:9 tão facilmente quanto para 5:4. As [figuras 1 a 4](#) trazem um passo-a-passo.

Listagem 3: Em busca de quadros entrelaçados

```
# transcode -V -i e4.mpv -J 32detect=verbose=1
...
(0) frame [000086]: (1) = 2141 | (2) = 2161 | (3) = 10 | interlaced = yes
...
```

A primeira coisa que precisamos fazer é corrigir a proporção. Precisamos expandir a imagem lateralmente por um fator de $(4/3)/(5/4)$ ou $(1.33/1.25 = 1.06)$. A imagem cresce em 6% e agora tem 768 pixels de largura. O próximo passo é mandar o transcode remover as barras negras no topo e na parte de baixo da tela, que ocupam 56 pixels cada; o resultado é uma imagem com a altura de 464 pixels.

Mandamos então o transcode reduzir a imagem de 768x464 pixels para uma largura de 640 pixels; é um fator de 1.2. A imagem resultante tem 640x386 pixels e, por isso, precisamos reduzir a altura para 384, o valor mais próximo divisível por 16. O fator de distorção introduzido é invisível para o olho humano. Algumas contas simples apontam o seguinte comando para converter uma gravação original de um DVR para um pequeno vídeo no formato *XviD*:

```
# transcode -V -i e4.mpv
-p c0.mpa
-o video_640.avi -y
xvid4 -w 1650
-j 56 -J smartyuv -Z
640x384,fast
```

A opção `-p` adiciona a trilha de áudio (as trilhas de áudio e vídeo estão armazenados em arquivos distintos) ao arquivo. A taxa de bits do vídeo é de 1650 bits por segundo, que combinada com a resolução selecionada dá um valor BPP de 0,267, o que é uma qualidade alta o suficiente para nossos fins.

Flexível e poderoso

Este artigo apenas roça a superfície do grande conjunto de recursos oferecidos pelo *transcode*. O programa vem também com uma coleção de práticas ferramentas de linha de comando, apresentadas na **tabela 5**. Consulte as listas de discussão do *transcode* [3]; elas são uma útil fonte de informações sobre o programa. ■

SOBRE O AUTOR

Jörn Reder é um entusiasta do código aberto e um ávido desenvolvedor de Perl. Ganhou experiência com o *transcode* enquanto programava o front-end *dvd::rip*, para extração e conversão de DVDs. Jörn trabalha para a *dimeDIS* em Colônia, na Alemanha, onde está envolvido principalmente no desenvolvimento de aplicativos de banco de dados para Internet.

INFORMAÇÕES

- [1] Homepage do *transcode*: <http://www.transcoding.org/>
- [2] O *VDRsync* faz o pré-processamento de gravações VDR para manipulação com o *transcode*: <http://vdrsync.vdr-portal.de/>
- [3] Listas de discussão sobre o *transcode*: <http://lists.exit1.org/mailman/listinfo/>
- [4] *Video Help*, site com muita informação sobre a autoria de VCDs, SVCDs e DVDs: <http://www.videohelp.com/>
- [5] *mencoder*, uma alternativa ao *transcode* baseada no *mplayer*. Inclui utilitário para criação de VCDs: <http://www.mplayerhq.hu/>
- [6] *dvd::rip*, interface para o *transcode* e outras ferramentas para extração de DVDs: <http://www.exit1.org/dvdrip/>

Tabela 5: Ferramentas de Linha de Comando do *transcode*

Nome	Descrição
<code>tccat</code>	Lê arquivos de vídeo e os exibe através de uma saída padrão. Normalmente é empregado internamente pelo <i>transcode</i> , mas pode ser usado para ler títulos de DVD.
<code>tcscan</code>	Analisa um arquivo de vídeo e devolve informações sobre ele. Tipicamente usado para encontrar o volume máximo numa faixa de áudio.
<code>tcmoinfo</code>	Exibe a documentação de um módulo ou filtro do <i>transcode</i> .
<code>tcprobe</code>	Analisa um arquivo de vídeo e devolve suas principais características técnicas. Rode uma análise com o <i>tcprobe</i> antes de processar com o <i>transcode</i> para certificar-se de ter as configurações corretas.
<code>tcquant</code>	Requantizador de MPEG. Reduz a taxa de bits de um arquivo MPEG sem recodificá-lo. Produz resultados de alta qualidade em pouco tempo.
<code>tcplex</code>	Multiplexação de trilhas elementares no formato MPEG (áudio + vídeo) para montar um programa completo.
<code>avimerge</code>	Funde diversos arquivos AVI para formar um só.
<code>avisplit</code>	Divide um arquivo AVI em diversos pedaços menores.