

Кодирование видео при помощи GPU из файла, запись с экрана и стриминг

Драйвер Intel использует API под названием VA-API, драйвер Amdgpu - OpenMAX, NVIDIA - Nvenc. А драйверы fglrx и nouveau не имеют поддержки аппаратного энкодера.

1. Кодирование при помощи GPU (из файла или запись с экрана)

Несколько лет назад существовал проект Gears on Gallium, который проводил бенчмарки драйверов и игр (преимущественно опенсорсных, но бывали и исключения - например эти ребята были первыми, кто опубликовал результаты тестов Doom 2016 под Wine Staging + AMDGPU-PRO + Vulkan), формировал дистрибутив Linux в форме Docker-контейнера с последним графическим стеком (из GIT), и много чего ещё... К сожалению, [проект закрыт](#). Сайт проекта - оффлайн, но остался [YouTube-канал](#).

[Одним из их проектов](#) было создание патча для GStreamer, оптимизирующего "узкие места" при кодировании видео, значительно снижая нагрузку на CPU. Патч доступен [на GitHub](#).

Также по ссылке доступен скрипт rec.sh, являющейся простой графической оболочкой на KDialog/zenity. При помощи неё можно начать [скринкаст](#), задав следующие настройки: а). Через что кодировать (Software, VA-API (Intel), OpenMAX (AMD open source)) б). Полный экран или окно? в). Куда сохранять? В итоге формируется команда наподобие такой:

```
gst-launch-1.0 -e ximagesrc use-damage=0 ! queue ! video/x-raw,format=BGRx !
videoconvert ! video/x-raw,format=I420,framerate=30/1 ! queue ! vaapiencode_h264
! h264parse ! queue ! matroskamux ! progressreport ! filesink
location=/home/username/Videos/screencast_current_date.mkv
```

Хотя написано "1.0", на самом деле это приложение называется так же и в Gstreamer 1.2, и в 1.4, и т.д. Чтобы команда работала, в вашей системе должен быть установлен пакет gstreamer-plugins-vaapi.

Также на AMD и Intel можно кодировать при помощи mrv - вот [сравнение](#) производительности gst и mrv.

Кодировать на GPU от NVIDIA можно при помощи ffmpeg [начиная](#) с версии 2.6. Вот пример команды:

```
ffmpeg -f x11grab -r 30 -s 1920x1080 -i :0.0 -f alsa -i pulse -ac 2 -ar 48000
-acodec libfdk-aac -vcodec nvenc -preset hq -tune film -threads 0 output.mkv
```

Я использую параметры "-preset hq -tune film", узнать остальные можно, выполнив "ffmpeg -h encoder=nvenc"

2. Стриминг в Twitch

rec.sh не знает как это делать, поэтому вручную:

```
gst-launch-1.0 ximagesrc use-damage=0 ! video/x-raw,format=BGRx !
videoconvert ! video/x-raw,format=I420,framerate=30/1 ! queue leaky=downstream !
x264enc speed-preset=faster qp-min=30 tune=zerolatency ! queue ! flvmux
name=muxer pulsesrc device=alsa_output.pci-0000_00_1b.0.analog-stereo.monitor !
queue ! voaacenc bitrate=128000 ! aacparse ! queue ! muxer.muxer. ! rtmpsink
location="rtmp://live.justin.tv/app/live_BAW_ID"
```

Если не работает кодирование аудио, убедитесь что установлен

```
ffmpeg -f x11grab -s "1920x1080" -r "30" -i :0.0 -f alsa -i pulse -f flv -ac 2 -ar 44100 -vcodec nvenc -s 1280x720 -preset hq -tune film -acodec libmp3lame -threads 0 -strict normal "rtmp://live.justin.tv/app/live_BAШ_ID"
```

Также можно просто воспользоваться программным обеспечением под названием [OBS](#)

3. Как задействовать аппаратное кодирование в OpenShot, PiTiVi и других редакторах видео

Первый использует ffmpeg, второй - GStreamer. Важно чтобы ваш системный ffmpeg был скомпилирован с флагами "--enable-nonfree --enable-nvenc". И тогда в списке доступных энкодеров появится nvenc. Если и в этом случае он не появляется - попробуйте пересобрать libmlt - посредника между OpenShot и ffmpeg.

4. Возможные проблемы

У меня было так, что выбран не тот VA-API backend, и поэтому при запуске команды из консоли была ошибка "не удаётся кодировать при помощи fglrx_drv_video.so". Возможно что причина в том, что раньше у меня была видеокарточка от AMD. Решается так: export LIBVA_DRIVER_NAME=i965. Можно также добавить это в /etc/environment.

А вообще, если что-то работает не так то для отладки полезен параметр "export GST_DEBUG=4".

Если не работает стриминг при помощи GStreamer, убедитесь что установлен плагин gstreamer-plugins-bad. В openSUSE мне помогла установка пакета gstreamer-plugins-bad-orig-addon.

Для ffmpeg существует [патч nvresize](#), позволяющий кодировать один видеопоток сразу в несколько файлов: 240p, 360p, 480p, 720p и 1080p. Патч использует CUDA, тогда как основной функционал nvenc использует только аппаратный энкодер. [Патч](#) применим только к FFmpeg 3.0 и 3.1, а в версии 3.2 он был [заменён](#) на новую технологию CUVID, умеющую гораздо больше возможностей, и также задействующую CUDA.