

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.



Learn, Share, Build

Each month, over 50 million developers come to Stack Overflow to learn, share their knowledge, and build their careers.

Join the world's largest developer community.

Google

Facebook

OR

How to use clCreateProgramWithBinary in OpenCL?



Tired of recruiter spam?
Want jobs tailored to your needs?



Get started

I'm trying to just get a basic program to work using `clCreateProgramWithBinary`. This is so I know how to use it rather than a "true" application.

I see that one of the parameters is a list of binaries. How exactly would I go about creating a binary to test with? I have some test code which creates a program from source, builds and enqueues it. Is there a binary created at some point during this process which I can feed into `clCreateProgramWithBinary`?

Here is some of my code, just to give an idea of my overall flow. I've omitted comments and error checks for simplicity.

```
program = clCreateProgramWithSource(clctx, 1, &dumbkernelsource, NULL, &errcode);  
errcode = clBuildProgram(program, env->num_devices, env->device, NULL, NULL, NULL);  
mykernel = clCreateKernel(program, "flops", &errcode);  
errcode = clGetKernelWorkGroupInfo(mykernel, *(env->device),  
    CL_KERNEL_WORK_GROUP_SIZE, sizeof(local), &local, NULL);
```

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
NULL, NULL);
```

opengl

edited Apr 9 at 7:59



Ciro Santilli 刘晓波死 六
四事件 法轮功

86.4k 15 349 287

asked Sep 7 '11 at 18:27



Adam S

3,381 12 43 83

3 Answers

After you compile your program, you can get its binary code with [clGetProgramInfo](#), and then save it to a file.

Example code (not tried to compile, but should be something along these lines):

```
program = clCreateProgramWithSource(clctx, 1, &dumbkernelsource, NULL, &errcode);
errcode = clBuildProgram(program, env->num_devices, env->device, NULL, NULL, NULL);
int number_of_binaries;
char **binary;
int *binary_sizes;
errcode = clGetProgramInfo(program, CL_PROGRAM_BINARY_SIZES, NULL, 0,
&number_of_binaries);
binary_sizes = new int[number_of_binaries];
binary = new char*[number_of_binaries];
errcode = clGetProgramInfo(program, CL_PROGRAM_BINARY_SIZES, binary_sizes,
number_of_binaries*sizeof(int), &number_of_binaries);
for (int i = 0; i < number_of_binaries; ++i) binary[i] = new char[binary_sizes[i]];
errcode = clGetProgramInfo(program, CL_PROGRAM_BINARIES, binary,
number_of_binaries*sizeof(char*), &number_of_binaries);
```

edited Sep 8 '11 at 8:06

answered Sep 7 '11 at 18:45



aland

3,500 2 13 31

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.



The official OpenCL Programming Guide book has a nice example of this. There's also a Google code project, [openc1-book-samples](#), which includes the code from the book. The example you're looking for is [here](#).

answered Apr 16 '13 at 12:59



[user2023370](#)

3,996 1 24 55

Minimal runnable example

Compile the embedded vector increment shader from CL C source, save the binary to `a.bin`, load the binary shader, and run it:

```
./a.out
```

Assertions are done at the end of the program.

Ignore the CL C shader, load binary from `a.bin`, and run it:

```
./a.out 0
```

Compile and run with:

```
gcc -ggdb3 -std=c99 -Wall -Wextra a.c -lOpenCL && ./a.out
```

Tested in Ubuntu 16.10, NVIDIA NVS5400, driver 375.39.

GitHub upstream: https://github.com/cirosantilli/cpp-cheat/blob/b1e9696cb18a12c4a41e0287695a2a6591b04597/openc1/binary_shader.c

```
#include <assert.h>
#include <stdio.h>
```

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
#define CL_USE_DEPRECATED_OPENCL_1_2_APIS
#include <CL/cl.h>

const char *source =
    "__kernel void kmain(__global int *out) {\n"
    "    out[get_global_id(0)]++; \n"
    "}\n"
;

#define BIN_PATH "a.bin"

char* common_read_file(const char *path, long *length_out) {
    char *buffer;
    FILE *f;
    long length;

    f = fopen(path, "r");
    assert(NULL != f);
    fseek(f, 0, SEEK_END);
    length = ftell(f);
    fseek(f, 0, SEEK_SET);
    buffer = malloc(length);
    if (fread(buffer, 1, length, f) < (size_t)length) {
        return NULL;
    }
    fclose(f);
    if (NULL != length_out) {
        *length_out = length;
    }
    return buffer;
}

int main(int argc, char **argv) {
    FILE *f;
    char *binary;
    cl_command_queue command_queue;
    cl_context context;
    cl_device_id device;
    cl_int input[] = {1, 2}, errcode_ret, binary_status;
    cl_kernel kernel, binary_kernel;
    cl_mem buffer;
    cl_platform_id platform;
    cl_program program, binary_program;
    const size_t global_work_size = sizeof(input) / sizeof(input[0]);
    int use_cache;
    long length;
```

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```

if (argc > 1) {
    use_cache = !strcmp(argv[1], "0");
} else {
    use_cache = 0;
}

/* Get the binary, and create a kernel with it. */
clGetPlatformIDs(1, &platform, NULL);
clGetDeviceIDs(platform, CL_DEVICE_TYPE_ALL, 1, &device, NULL);
context = clCreateContext(NULL, 1, &device, NULL, NULL, NULL);
command_queue = clCreateCommandQueue(context, device, 0, NULL);
if (use_cache) {
    binary = common_read_file(BIN_PATH, &lenght);
    binary_size = lenght;
} else {
    program = clCreateProgramWithSource(context, 1, &source, NULL, NULL);
    clBuildProgram(program, 1, &device, "", NULL, NULL);
    kernel = clCreateKernel(program, "kmain", NULL);
    clGetProgramInfo(program, CL_PROGRAM_BINARY_SIZES, sizeof(size_t),
&binary_size, NULL);
    binary = malloc(binary_size);
    clGetProgramInfo(program, CL_PROGRAM_BINARIES, binary_size, &binary, NULL);
    f = fopen(BIN_PATH, "w");
    fwrite(binary, binary_size, 1, f);
    fclose(f);
}
binary_program = clCreateProgramWithBinary(
    context, 1, &device, &binary_size,
    (const unsigned char *)&binary, &binary_status, &errcode_ret
);
free(binary);
clBuildProgram(binary_program, 1, &device, NULL, NULL, NULL);
binary_kernel = clCreateKernel(binary_program, "kmain", &errcode_ret);

/* Run the kernel created from the binary. */
buffer = clCreateBuffer(context, CL_MEM_READ_WRITE | CL_MEM_COPY_HOST_PTR,
sizeof(input), input, NULL);
clSetKernelArg(binary_kernel, 0, sizeof(buffer), &buffer);
clEnqueueNDRangeKernel(command_queue, binary_kernel, 1, NULL,
&global_work_size, NULL, 0, NULL, NULL);
clFlush(command_queue);
clFinish(command_queue);
clEnqueueReadBuffer(command_queue, buffer, CL_TRUE, 0, sizeof(input), input, 0,
NULL, NULL);

/* Assertions. */
assert(input[0] == 2);

```

Stack Overflow requires external JavaScript from another domain, which is blocked or failed to load.

```
/* Cleanup. */
clReleaseMemObject(buffer);
clReleaseKernel(kernel);
clReleaseKernel(binary_kernel);
clReleaseProgram(program);
clReleaseProgram(binary_program);
clReleaseCommandQueue(command_queue);
clReleaseContext(context);
return EXIT_SUCCESS;
}
```

I highly recommend `cat a.bin`, which contains human readable (and editable) PTX assembly for this implementation.

answered Apr 8 at 10:13



Ciro Santilli 刘晓波死 六
四事件 法轮功

86.4k 15 349 287