```
  return 0;
}

#ifdef __cplusplus          ← 为 extern "C"块添加右大括号
}
#endif
```

至此我们将代码修改为了其他模块可以调用，现在是时候执行第二步（参见图 7-6），并将代码编译为 WebAssembly 副模块了。

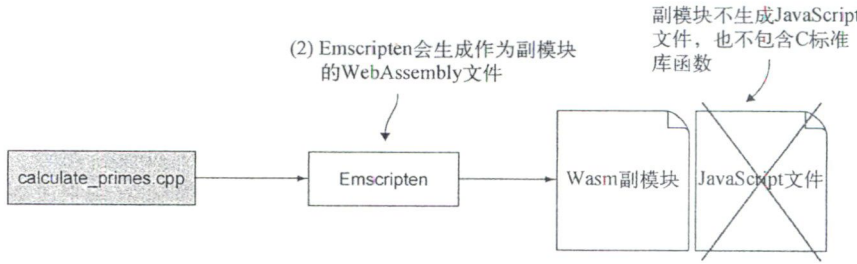



图 7-6　用 Emscripten 生成作为副模块的 WebAssembly 文件

### 2. 用 Emscripten 从 calculate_primes.cpp 生成作为副模块的 WebAssembly 文件

在前面的章节中创建 WebAssembly 副模块时，我们用第 4 章中创建的一些替代代码来代替 C 标准库函数。在不能使用 C 标准库函数的情况下，这样使得副模块仍然可以工作。本示例中不需要替代代码，因为副模块会在运行时链接到主模块，而主模块有 C 标准库函数。

要想将修改后的文件 calculate_primes.cpp 编译为 WebAssembly 副模块，需要打开命令行窗口，进入目录 Chapter 7\7.2.2 dlopen\source\，并运行以下命令。

```
emcc calculate_primes.cpp -s SIDE_MODULE=2 -O1
➥ -o calculate_primes.wasm
```

至此我们创建了副模块，下一步（参见图 7-7）是创建主模块。

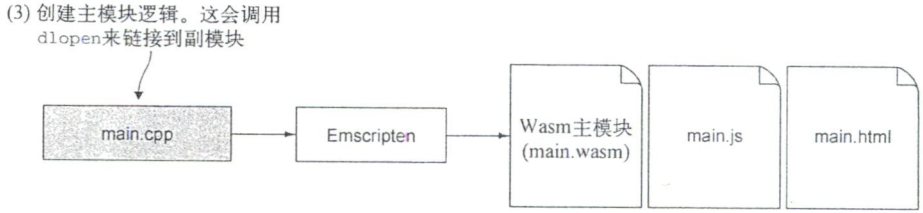



图 7-7　用 dlopen 实现动态链接的第三步是创建用 dlopen 链接到副模块的逻辑

### 3. 创建将要链接到副模块的逻辑

在目录 Chapter 7\7.2.2 dlopen\source\下，创建一个名为 main.cpp 的文件，然后用编辑器打开它。首先需要向文件 main.cpp 中添加头文件包含。在这个示例中，需要包含头文件 dlfcn.h（以