

BrainVM 规范

BrainVM 项目的灵感来源是号称“最小的图灵完备语言”Brainf*ck。Brain字节码的设计以Brainf*ck语言作为模板，可能会在未来添加新特性。

BrainVM 是自由软件，任何人可以将其应用于任何目的。

但强烈不建议应用到生产环境中，哈哈

1. Brain 字节码

在某种程度上说，Brain字节码是Brainf*ck语言的扩展。因此，这里提供一份两者的对照表。

Brain 字节码	描述	Brainf*ck
<code>leftmove n</code>	左移指针	<code><</code>
<code>rightmove n</code>	右移指针	<code>></code>
<code>add n</code>	给指针指向的值加n	<code>+</code>
<code>sub n</code>	给指针指向的值减n	<code>-</code>
<code>putchar</code>	将指针指向的值作为字符输出	<code>.</code>
<code>getchar</code>	输入一个字符，储存为指针指向的值	<code>,</code>
<code>while</code>	如果条件不为0，执行循环	<code>[</code>
<code>endwhile</code>	如果开始循环时的条件此时为0，结束循环	<code>]</code>
<code>putint</code>	将指针指向的值作为整数输出	无
<code>getint</code>	输入一个整数，储存为指针指向的值	无

注意: 一些指令后跟了一个数字 `n`，这表示您可以将指针一次移动多个位置（或一次将其所指向的值加减n），这与其前身Brainf*ck语言不同。

还有，`while` 和 `endwhile` 的定义与Brainf*ck语言中的稍有不同。BrainVM在开始循环时会储存下循环条件的地址，而Brainf*ck语言总是以当前所指向的值作为循环条件。

2. 虚拟机参数

BrainVM提供了一些有用的参数关于。在介绍这些参数之前，别忘了 `-version` 参数会打印出虚拟机的版本。其他参数如下所示：

1. 初始内存块大小

您可以指定BrainVM的初始内存块大小，当内存不足时，BrainVM会自动扩充内存。下面的指令

```
bvm -initial 1024
```

将BrainVM的初始内存块设置为1024。实际上，“内存块设置为1024”表示BrainVM内部有1024个int用作虚拟机的“内存”。这与Brainf*ck语言所定义的char有所不同。

2. 扩充阈值

当内存块不足时，BrainVM自动扩充其内存。下面的指令

```
bvm -threshold 1024
```

将扩充阈值设置为1024。这表示当内存不足时，如果内存块大小小于扩充阈值，就简单地将其扩充至原来的2倍；当内存块大小大于扩充阈值，就将其加上扩充阈值。

3. 翻译

由于Brain字节码是对Brainf*ck的简单扩充，可以很容易地将Brainf*ck语言转化为Brain字节码。下面的指令

```
bvm -translate hello.bf
```

将Brainf*ck文件 `hello.bf` 转化为Brain字节码文件 `hello.brain`。注意，由于Brainf*ck语言中 `[/]` 和Brain字节码 `while / endwhile` 的定义有所不同，从Brain字节码到Brainf*ck语言的转换很难实现。

4. 优化

通常来说，翻译过后的字节码文件中有很多低效的代码。使用下面的指令，您可以简单地优化它们：

```
bvm -optimize hello.bf
```

不仅是从Brainf*ck语言转换而来的字节码，您自己所撰写的字节码文件可能也能得到优化。

5. 速度测试

在编程领域中，速度测试一直很重要。指定此参数后，在程序运行结束之后，虚拟机将会打印出程序的执行时间。

```
bvm -speedtest hello.bf
```

这就是您所需要的参数。

一些参数可以搭配使用。

3. 示例

随此BrainVM一起发布的example文件夹中，含有BrainVM的可执行程序及一些示例代码（字节码）。

您可以首先运行

```
bvm -translate hello.bf
```

以测试BrainVM对Brainf*ck的翻译功能。

然后，可以执行

```
bvm -optimize hello.brain
```

以测试BrainVM对字节码的优化功能。

执行

```
bvm -speedtest hello.brain
```

以测试BrainVM对字节码的执行正确与否，并知晓其运行速度。

如有其他字节码文件，可以用相同的方法对其进行测试，这里不再赘述。

注意：以 `.bf` 结尾的文件是Brainfuck语言代码，需要转换成字节码才能正确执行。以 `.brain` 结尾的文件已经是（优化过的）Brain字节码，无需转换（执行转换操作则会失败）。

<https://blog.csdn.net/codebat/article/details/38585665>

这是Brainfuck语言的入门教程

<https://fatiherikli.github.io/brainfuck-visualizer/>

这个网页可以贴自己的BF代码执行，可以输入输出，还可以看到可视化的运行过程