

**DOT:**

\* dot 画一幅图主要有4个步骤。

\* 主要分为有向图digraph和graph无向图，其中有向图使用“->”无向图使用“--”。

**概念：**

dot接收dot语句作为输入，主要有3种对象概念：

**Node**、**graphs**、**edges**。

程序中graph是最外层的对象，主要分为有向图和无向图。

程序中node，表示设置点的默认值。

程序中edge，表示设置边的默认值。

给对象添加属性可以附在中括号“[]”中，**使用/\* \*/进行注释**。**//注释**

点的默认参数：**shape**=ellips，**width**=0.75，**height**=0.5 and **labeled** by the node name。

一些点的形状在appendix.h中，一些常用的shape形状有box，circle，record，plaintext。

graphviz的一些属性修改：

\* 属性是区分大小写的。

\*使用inch或者points来描述，1inch = 72points。

\*

**常用属性说明：**

**这些属性可以用在graph、node、edge上，但是意义不同。**

**\*【shape 】**形状：box(方形)、plaintext(无边框)、rectangle(圆角长方)、

record(都是一些格子port)

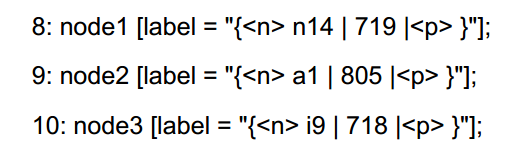
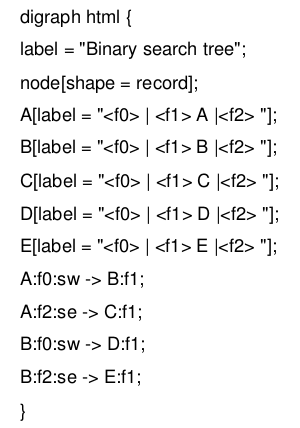
\*【**color**】颜色 ：red(红色)

也可以使用3色数值“.7 .3 1.0”

**\*【weight】**边重：默认为1；

**\*【style】**线类型：dotted(点状的线)、filled(填充的)

**shape：record**



**<f0>|<f1>貌似是竖的，按上下排列。**

**<n>|<p> 貌似是横的，按左右排列的。n/p是什么意思？?**

**错，是按大括号对{}，来进行横竖的。有大括号是横，无大括号是竖。**

**GRAPH：**

\*【rankdir 】图的布局方向：LR/ RL/ TB/ BT

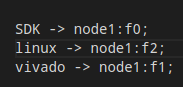
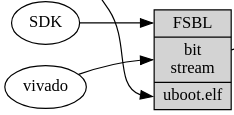
\*【subgraph 】：子图中的cluster是**必须有的**；

subgraph **cluster**#：

{ 可以设置子图的属性。}

**NODE：**

**\* record中的端口port，可以用于来指向别的 （node：port)**

 ****

\*【label】线边的注释：

“”：可以是双引号括起来的字符串“make a\n string”。

\n：格式控制。

\ ：转义，空格。

格子：如果shape定义为record，则可以使用<f#>|画格子，<n>|<p>表示左右。

node：一个node是标签，两个node是线上的注释，三个node位于2条线上注释。

\*【dir】线箭头方向：both、none、back、forward（默认）。

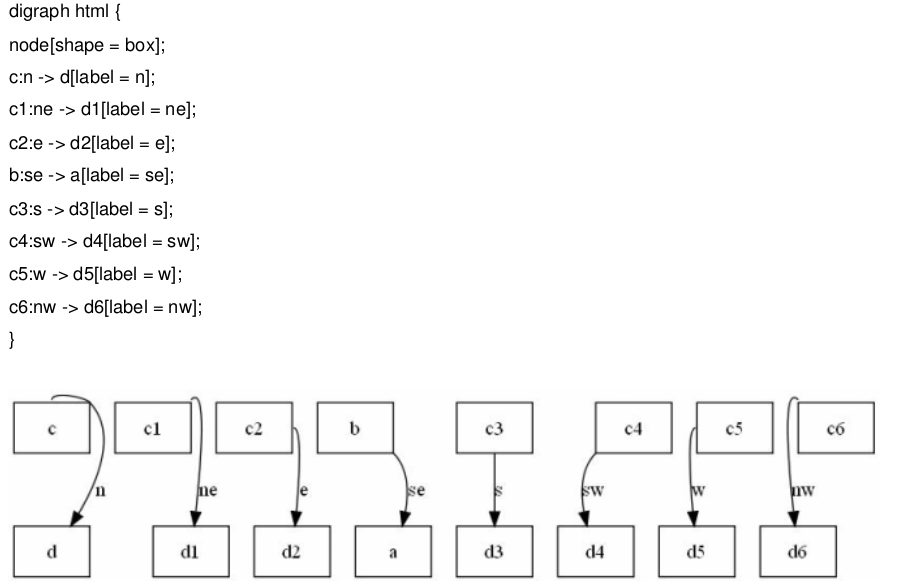
\* 连接多个node：

execute -> {a ; b}

**EDGE：**

\* node后面使用**“：”指定线的方向。**

设立一条边时, 我们可以制定这条边从起点的那个位置射出和从哪个位置结束。使用方位进行描述：东、南、西、北。e、s、w、n，es、sw。。。。。



================ 注意 ===============

1.程序中的单个节点名称node，不能用“空格”隔开，会被识别为2个node；

2.label出现在2个node之间的注释，如果是一个node的话，就是node名字。

3.layout的引擎engine有很多：dot、circo、fdp、sfdp、neato、nop、osage、patchwork