

Codeblock class: graphviz

```
sudo apt-get install graphviz
http://graphviz.org
```

runs:

```
> {im_prg} {im_opt} -T{im_fmt} <fname>.{im_prg} <fname>.{im_fmt}
```

class->cmd

```
dot -> dot
neato -> neato
twopi -> twopi
circo -> circo
fdp -> fdp
sfdp -> sfdp
graphviz -> dot
```

Metadata options

```
imagine.im_out: img, fcb
imagine.im_log: 4
imagine.fdp.im_fmt: svg
imagine.graphviz.im_fmt: svg
```

Notes

- graphviz class defaults to dot

graphviz examples

Graphviz (svg)

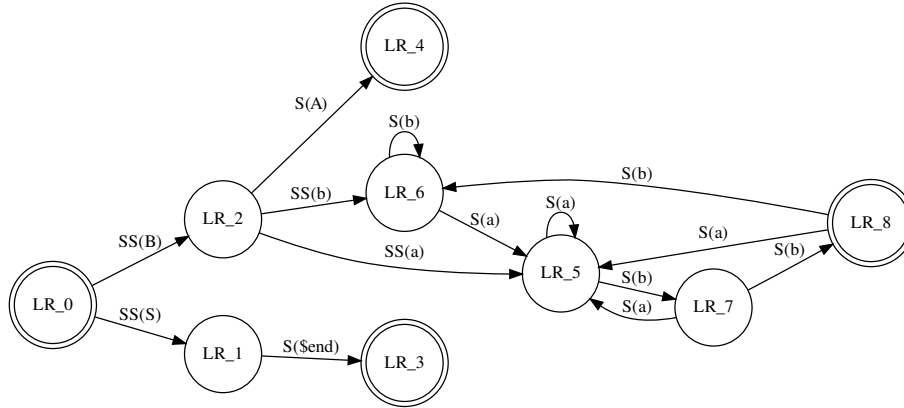


Figure 1: Created by Graphviz

```

```{.graphviz caption="Created by Graphviz"}

digraph finite_state_machine {
 rankdir=LR;
 node [shape = doublecircle]; LR_0 LR_3 LR_4 LR_8;
 node [shape = circle];
 LR_0 -> LR_2 [label = "SS(B)"];
 LR_0 -> LR_1 [label = "SS(S)"];
 LR_1 -> LR_3 [label = "S($end)"];
 LR_2 -> LR_6 [label = "SS(b)"];
 LR_2 -> LR_5 [label = "SS(a)"];
 LR_2 -> LR_4 [label = "S(A)"];
 LR_5 -> LR_7 [label = "S(b)"];
 LR_5 -> LR_5 [label = "S(a)"];
 LR_6 -> LR_6 [label = "S(b)"];
 LR_6 -> LR_5 [label = "S(a)"];
 LR_7 -> LR_8 [label = "S(b)"];
 LR_7 -> LR_5 [label = "S(a)"];
 LR_8 -> LR_6 [label = "S(b)"];
 LR_8 -> LR_5 [label = "S(a)"];
}
```

```

fdp

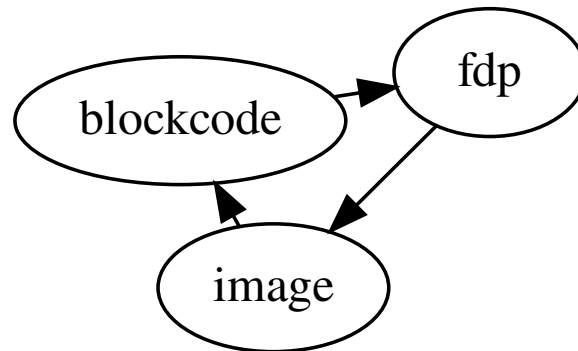


Figure 2: Created by fdp

```
```{.fdp caption="Created by fdp"}
```

```
digraph {
 blockcode -> fdp;
 fdp -> image;
 image -> blockcode;
}
```

```
```
```

sfdp

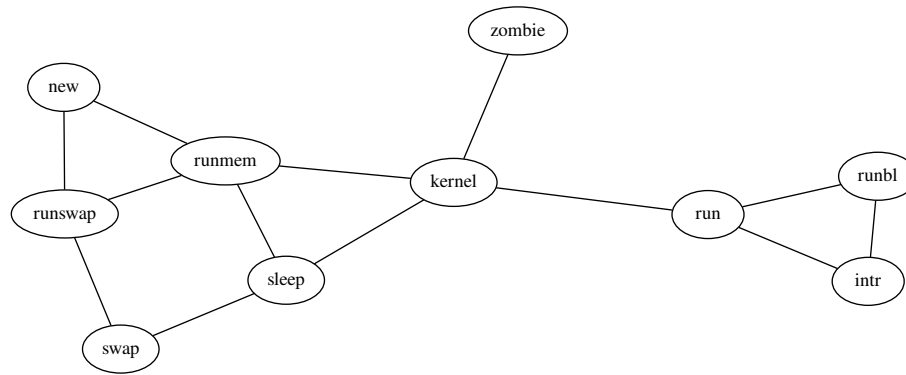


Figure 3: Created by sfdp

```

```{.sfdp caption="Created by sfdp"}
graph G {
run -- intr;
intr -- runbl;
runbl -- run;
run -- kernel;
kernel -- zombie;
kernel -- sleep;
kernel -- runmem;
sleep -- swap;
swap -- runswap;
runswap -- new;
runswap -- runmem;
new -- runmem;
sleep -- runmem;
}
```

```

neato

States in a kernel OS plotted by neato:

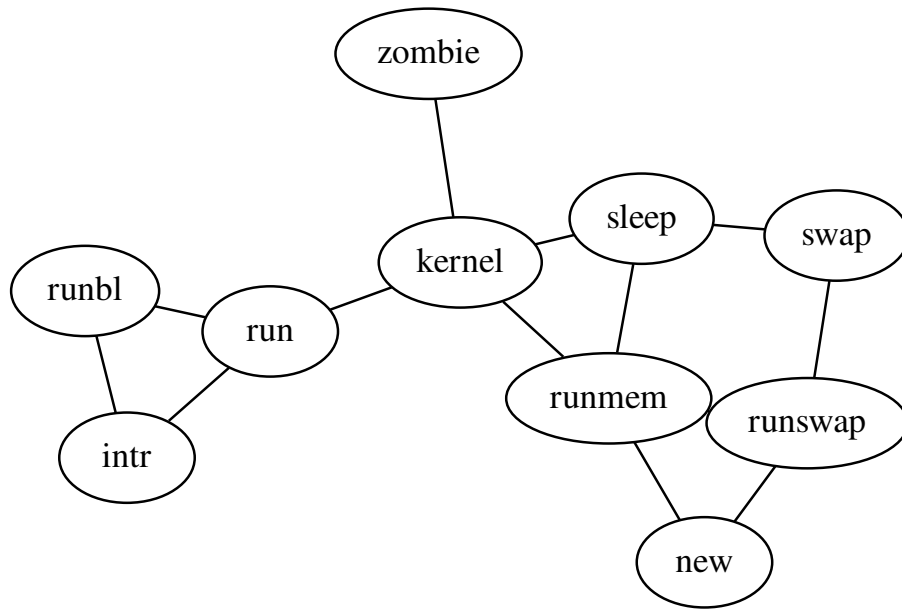


Figure 4: Created by neato

```
```.neato caption="Created by neato"
graph G {
 run -- intr;
 intr -- runbl;
 runbl -- run;
 run -- kernel;
 kernel -- zombie;
 kernel -- sleep;
 kernel -- runmem;
 sleep -- swap;
 swap -- runswap;
 runswap -- new;
 new -- runmem;
 runswap -- runmem;
 sleep -- runmem;
}
...

```

twopi

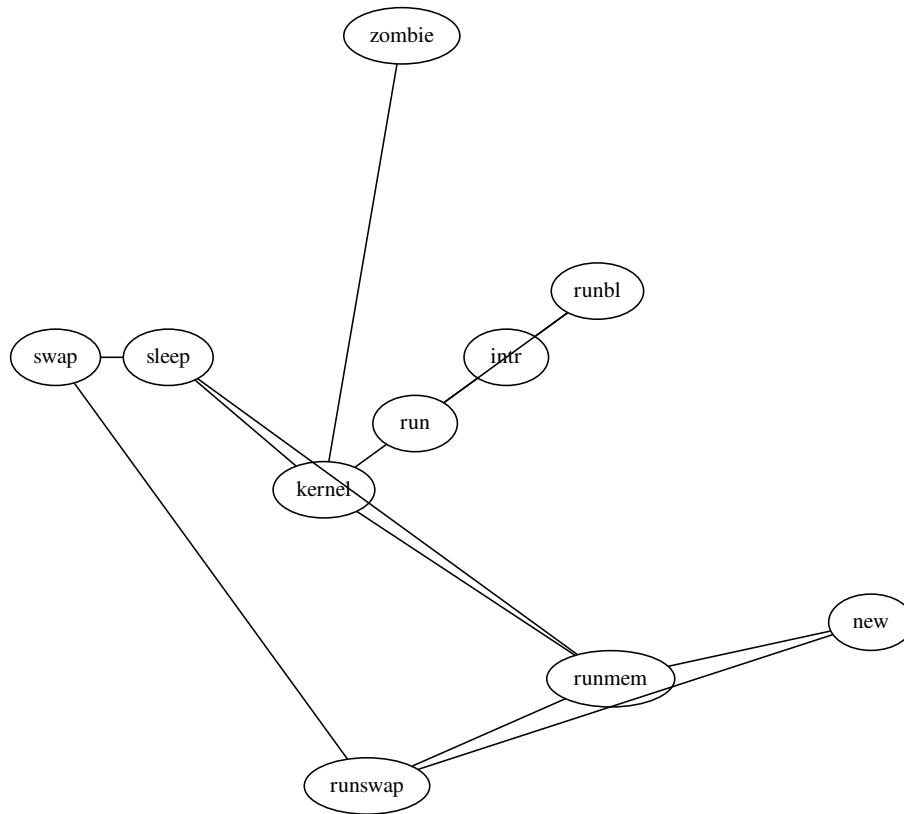


Figure 5: Created by twopi

```

```{.twopi caption="Created by twopi"}
graph G {
run -- intr;
intr -- runbl;
runbl -- run;
run -- kernel;
kernel -- zombie;
kernel -- sleep;
kernel -- runmem;
sleep -- swap;
swap -- runswap;
runswap -- new;
runswap -- runmem;
new -- runmem;
sleep -- runmem;
}

```

}

circo

Again, the same but by **circo**:

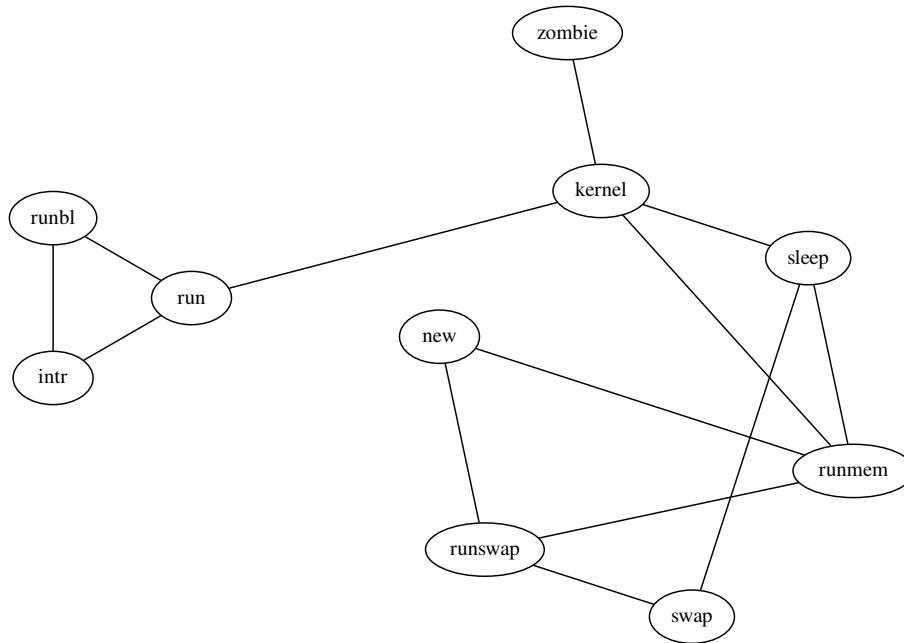


Figure 6: created by circo

```
```{.circo caption="created by circo"}
```

```
graph G {
run -- intr;
intr -- runbl;
runbl -- run;
run -- kernel;
kernel -- zombie;
kernel -- sleep;
kernel -- runmem;
sleep -- swap;
swap -- runswap;
runswap -- new;
runswap -- runmem;
new -- runmem;
sleep -- runmem;
}
```
```


Documentation

dot -h

Usage: dot [-Vv?] [-(GNE)name=val] [-(KtIso)<val>] <dot files>
(additional options for neato) [-x] [-n<v>]
(additional options for fdp) [-L(g0)] [-L(nUCT)<val>]
(additional options for memtest) [-m<v>]
(additional options for config) [-cv]

-V - Print version and exit
-v - Enable verbose mode
-Gname=val - Set graph attribute 'name' to 'val'
-Nname=val - Set node attribute 'name' to 'val'
-Ename=val - Set edge attribute 'name' to 'val'
-Tv - Set output format to 'v'
-Kv - Set layout engine to 'v' (overrides default based on command name)
-lv - Use external library 'v'
-ofile - Write output to 'file'
-O - Automatically generate an output filename based on the input filename with a
-P - Internally generate a graph of the current plugins.
-q[l] - Set level of message suppression (=1)
-s[v] - Scale input by 'v' (=72)
-y - Invert y coordinate in output

-n[v] - No layout mode 'v' (=1)
-x - Reduce graph

-Lg - Don't use grid
-LO - Use old attractive force
-Ln<i> - Set number of iterations to i
-LU<i> - Set unscaled factor to i
-LC<v> - Set overlap expansion factor to v
-LT[*]<v> - Set temperature (temperature factor) to v

-m - Memory test (Observe no growth with top. Kill when done.)
-m[v] - Memory test - v iterations.

-c - Configure plugins (Writes \$prefix/lib/graphviz/config
 with available plugin information. Needs write privilege.)
-? - Print usage and exit

man page

GRAPHVIZ(7)

Miscellaneous Information Manual

GRAPHVIZ(7)

NAME

graphviz - rich set of graph drawing tools

SYNOPSIS

This manpage has been written to fulfil the need of a centralized documentation presenting all available tools in the graphviz package.

AVAILABLE TOOLS

Graph layout programs

dot filter for hierarchical layouts of graphs

neato filter for symmetric layouts of graphs

twopi filter for radial layouts of graphs

circo filter for circular layout of graphs

fdp filter for symmetric layouts of graphs

All of the filters work with either directed or undirected graphs, though dot is typically used for directed graphs and neato for undirected graphs. Note also that neato -n[2] can be used to render layouts produced by the other filters.

Graph drawing programs

lefty A Programmable Graphics Editor

lneato lefty + neato

dotty lefty + dot

Graph layout enhancement

gvcolor

flow colors through a ranked digraph

unflatten

adjust directed graphs to improve layout aspect ratio

gvpack merge and pack disjoint graphs

Graph information and transformation

gc count graph components

acyclic

make directed graph acyclic

`nop` pretty-print graph file

`ccomps` connected components filter for graphs

`sccmap` extract strongly connected components of directed graphs

`tred` transitive reduction filter for directed graphs

`dijkstra`
 single-source distance filter

`bcomps` biconnected components filter for graphs

`gvpr` graph pattern scanning and processing language

`prune` prune directed graphs

Other

`gxl2dot`, `dot2gxl`
 GXL-DOT converters

AUTHOR

This manual page was written by Cyril Brulebois <cyril.brulebois@enst-bretagne.fr> in november 2006, based on an initial documentation effort by Joachim Berdal Haga <jbh@lupus.ig3.net>. It can be distributed under the same terms as the graphviz package.

November 19, 2006

GRAPHVIZ(7)