
```

% Flowcharting techniques for easy maintenance
% Author: Brent Longborough
\documentclass[x11names]{article}
\usepackage{tikz}
\usetikzlibrary{shapes,arrows,chains}
%%%<
\usepackage{verbatim}
\usepackage[active,tightpage]{preview}
\PreviewEnvironment{tikzpicture}
\setlength\PreviewBorder{5mm}%
%%%>
\begin{comment}
:Title: Easy-maintenance flowchart
:Tags: flowcharts
:Author: Brent Longborough
:Slug: flexible-flow-chart

This TikZ example illustrates a number of techniques for making TikZ
flowcharts easier to maintain:
* Use of <on chain> and <on grid> to simplify positioning
* Use of global <node distance> options to eliminate the need to
  specify individual inter-node distances
* Use of <join> to reduce the need for references to node names
* Use of <join by> styles to tailor specific connectors
* Use of <coordinate> nodes to provide consistent layout for
  parallel flow lines
* A method for consistent annotation of decision box exits
* A technique for marking coordinate nodes (for layout debugging)

I encourage you to tinker at this file - add intermediate boxes,
alter the global distance settings, and so on, to see how well (or
ill!) it adapts.
\end{comment}
\begin{document}
% =====
% Set up a few colours
\colorlet{lcfree}{Green3}
\colorlet{lcnorm}{Blue3}
\colorlet{lccong}{Red3}
% -----
% Set up a new layer for the debugging marks, and make sure it is on
% top
\pgfdeclarelayer{marx}
\pgfsetlayers{main,marx}
% A macro for marking coordinates (specific to the coordinate naming
% scheme used here). Swap the following 2 definitions to deactivate
% marks.
\providecommand{\cmark}[2][\{%
  \begin{pgfonlayer}{marx}
    \node [nmark] at (c#2#1) {#2};
  \end{pgfonlayer}{marx}
}
\providecommand{\cmark}[2][\relax}
% -----
% Start the picture
\begin{tikzpicture}[%
  >=triangle 60,           % Nice arrows; your taste may be different
  start chain=going below,  % General flow is top-to-bottom
  node distance=6mm and 60mm, % Global setup of box spacing
  every join/.style={norm}, % Default linetype for connecting boxes
]
% -----
% A few box styles

```

```

% <on chain> *and* <on grid> reduce the need for manual relative
% positioning of nodes
\tikzset{
  base/.style={draw, on chain, on grid, align=center, minimum height=4ex},
  proc/.style={base, rectangle, text width=8em},
  test/.style={base, diamond, aspect=2, text width=5em},
  term/.style={proc, rounded corners},
  % coord node style is used for placing corners of connecting lines
  coord/.style={coordinate, on chain, on grid, node distance=6mm and 25mm},
  % nmark node style is used for coordinate debugging marks
  nmark/.style={draw, cyan, circle, font={\sffamily\bfseries}},
  % -----
  % Connector line styles for different parts of the diagram
  norm/.style={->, draw, lcnorm},
  free/.style={->, draw, lcfree},
  cong/.style={->, draw, lccong},
  it/.style={font={\small\itshape}}
}
% -----
% Start by placing the nodes
\node [proc, densely dotted, it] (p0) {New trigger message thread};
% Use join to connect a node to the previous one
\node [term, join] {Trigger scheduler};
\node [proc, join] (p1) {Get quota  $k > 1$ };
\node [proc, join] {Open queue};
\node [proc, join] {Dispatch message};
\node [test, join] (t1) {Got msg?};
% No join for exits from test nodes - connections have more complex
% requirements
% We continue until all the blocks are positioned
\node [proc] (p2) { $k \leq 1$ };
\node [proc, join] (p3) {Dispatch message};
\node [test, join] (t2) {Got msg?};
\node [test] (t3) {Capacity?};
\node [test] (t4) { $k \leq 1$ };
% We position the next block explicitly as the first block in the
% second column. The chain 'comes along with us'. The distance
% between columns has already been defined, so we don't need to
% specify it.
\node [proc, fill=lcfree!25, right=of p1] (p4) {Reset congestion};
\node [proc, join=by free] {Set \textsc{mq} wait flag};
\node [proc, join=by free] (p5) {Dispatch message};
\node [test, join=by free] (t5) {Got msg?};
\node [test] (t6) {Capacity?};
% Some more nodes specifically positioned (we could have avoided this,
% but try it and you'll see the result is ugly).
\node [test] (t7) [right=of t2] { $k \leq 1$ };
\node [proc, fill=lccong!25, right=of t3] (p8) {Set congestion};
\node [proc, join=by cong, right=of t4] (p9) {Close queue};
\node [term, join] (p10) {Exit trigger message thread};
% -----
% Now we place the coordinate nodes for the connectors with angles, or
% with annotations. We also mark them for debugging.
\node [coord, right=of t1] (c1) {}; \cmark{1}
\node [coord, right=of t3] (c3) {}; \cmark{3}
\node [coord, right=of t6] (c6) {}; \cmark{6}
\node [coord, right=of t7] (c7) {}; \cmark{7}
\node [coord, left=of t4] (c4) {}; \cmark{4}
\node [coord, right=of t4] (c4r) {}; \cmark[r]{4}
\node [coord, left=of t7] (c5) {}; \cmark{5}
% -----
% A couple of boxes have annotations
\node [above=0mm of p4, it] {(Queue was empty)};

```

```

\node [above=0mm of p8, it] {(Queue was not empty)};
% -----
% All the other connections come out of tests and need annotating
% First, the straight north-south connections. In each case, we first
% draw a path with a (consistently positioned) annotation node, then
% we draw the arrow itself.
\path (t1.south) to node [near start, xshift=1em] {$y$} (p2);
\draw [*->,lcnorm] (t1.south) -- (p2);
\path (t2.south) to node [near start, xshift=1em] {$y$} (t3);
\draw [*->,lcnorm] (t2.south) -- (t3);
\path (t3.south) to node [near start, xshift=1em] {$y$} (t4);
\draw [*->,lcnorm] (t3.south) -- (t4);
\path (t5.south) to node [near start, xshift=1em] {$y$} (t6);
\draw [*->,lcfree] (t5.south) -- (t6);
\path (t6.south) to node [near start, xshift=1em] {$y$} (t7);
\draw [*->,lcfree] (t6.south) -- (t7);
% -----
% Now the straight east-west connections. To provide consistent
% positioning of the test exit annotations, we have positioned
% coordinates for the vertical part of the connectors. The annotation
% text is positioned on a path to the coordinate, and then the whole
% connector is drawn to its destination box.
\path (t3.east) to node [near start, yshift=1em] {$n$} (c3);
\draw [o->,lccong] (t3.east) -- (p8);
\path (t4.east) to node [yshift=-1em] {$k \leq 0$} (c4r);
\draw [o->,lcnorm] (t4.east) -- (p9);
% -----
% Finally, the twisty connectors. Again, we place the annotation
% first, then draw the connector
\path (t1.east) to node [near start, yshift=1em] {$n$} (c1);
\draw [o->,lcfree] (t1.east) -- (c1) |- (p4);
\path (t2.east) -| node [very near start, yshift=1em] {$n$} (c1);
\draw [o->,lcfree] (t2.east) -| (c1);
\path (t4.west) to node [yshift=-1em] {$k > 0$} (c4);
\draw [*->,lcnorm] (t4.west) -- (c4) |- (p3);
\path (t5.east) -| node [very near start, yshift=1em] {$n$} (c6);
\draw [o->,lcfree] (t5.east) -| (c6);
\path (t6.east) to node [near start, yshift=1em] {$n$} (c6);
\draw [o->,lcfree] (t6.east) -| (c7);
\path (t7.east) to node [yshift=-1em] {$k \leq 0$} (c7);
\draw [o->,lcfree] (t7.east) -- (c7) |- (p9);
\path (t7.west) to node [yshift=-1em] {$k > 0$} (c5);
\draw [*->,lcfree] (t7.west) -- (c5) |- (p5);
% -----
% A last flourish which breaks all the rules
\draw [->,MediumPurple4, dotted, thick, shorten >=1mm]
(p9.south) -- ++(5mm,-3mm) -- ++(27mm,0)
|- node [black, near end, yshift=0.75em, it]
{(When message + resources available)} (p0);
% -----
\end{tikzpicture}
% =====
\end{document}

```