```perl
 1: #!/usr/bin/perl
 2: # $Id: graph.perl,v 1.1 2018-11-06 18:50:43-08 - - $
 3:
 4: use strict;
 5: use warnings;
 6: $0 =~ s|.*/||;
 7:
 8: # Example setting up a directed graph.
 9:
10: my @inputs = (
11:     "all : hello",
12:     "hello : main.o hello.o",
13:     "main.o : main.c hello.h",
14:     "hello.o : hello.c hello.h",
15:     "ci : Makefile main.c hello.c hello.h",
16:     "test : hello",
17:     "clean : ",
18:     "spotless : clean",
19: );
20:
21: sub parse_dep ($) {
22:     my ($line) = @_;
23:     return undef unless $line =~ m/^(\S+)\s*:\s*(.*?)\s*$/;
24:     my ($target, $dependency) = ($1, $2);
25:     my @dependencies = split m/\s+/, $dependency;
26:     return $target, \@dependencies;
27: }
28:
29: my %graph;
30: for my $input (@inputs) {
31:     my ($target, $deps) = parse_dep $input;
32:     print "$0: syntax error: $input\n" and next unless defined $target;
33:     $graph{$target} = $deps;
34: }
35:
36: for my $target (keys %graph) {
37:     print "\"$target\"";
38:     my $deps = $graph{$target};
39:     if (not @$deps) {
40:         print " has no dependencies";
41:     }else {
42:         print " depends on";
43:         print " \"$_\"" for @$deps;
44:     }
45:     print "\n";
46: }
47:
```

```
1: "test" depends on "hello"
2: "clean" has no dependencies
3: "all" depends on "hello"
4: "main.o" depends on "main.c" "hello.h"
5: "ci" depends on "Makefile" "main.c" "hello.c" "hello.h"
6: "hello.o" depends on "hello.c" "hello.h"
7: "spotless" depends on "clean"
8: "hello" depends on "main.o" "hello.o"
```

```perl
 1: #!/usr/bin/perl
 2: # $Id: mtime.perl,v 1.1 2018-11-06 18:50:44-08 - - $
 3: #
 4: # NAME
 5: #    older.perl - check whether a pair of files are older or newer
 6: #
 7: # SYNOPSIS
 8: #    older.perl filename...
 9: #
10: # DESCRIPTION
11: #    The two files' modification times are compared and a
12: #    relationship is printed.
13: #
14:
15: use strict;
16: use warnings;
17: use POSIX qw(strftime);
18: $0 =~ s|.*/||;
19:
20: sub mtime ($) {
21:    my ($filename) = @_;
22:    my @stat = stat $filename;
23:    return @stat ? $stat[9] : undef;
24: }
25:
26: sub fileinfo ($) {
27:    my ($filename) = @_;
28:    my $mtime = mtime $filename;
29:    print "$filename: ";
30:    if (defined $mtime) {print strftime "%c\n", localtime $mtime}
31:                else {print "$!\n"}
32:    return $mtime;
33: }
34:
35: for my $filename (@ARGV) {
36:    unless (-e $filename) {
37:       printf STDERR "$0: $filename: $!\n";
38:    }else {
39:       my $mtime = mtime $filename;
40:       my $ctime = strftime "%c", localtime $mtime;
41:       printf "%-20s %12d %s\n", $filename, $mtime, $ctime;
42:    }
43: }
44:
```

```
 1: Listing.pdf            1541559045 Tue 06 Nov 2018 06:50:45 PM PST
 2: Listing.ps             1541559044 Tue 06 Nov 2018 06:50:44 PM PST
 3: RCS                    1541559136 Tue 06 Nov 2018 06:52:16 PM PST
 4: graph.output           1541559136 Tue 06 Nov 2018 06:52:16 PM PST
 5: graph.perl             1541559136 Tue 06 Nov 2018 06:52:16 PM PST
 6: mk                     1541559136 Tue 06 Nov 2018 06:52:16 PM PST
 7: mtime.output           1541559136 Tue 06 Nov 2018 06:52:16 PM PST
 8: mtime.perl             1541559136 Tue 06 Nov 2018 06:52:16 PM PST
 9: sigtoperl              1541559044 Tue 06 Nov 2018 06:50:44 PM PST
10: sigtoperl.cpp          1541559044 Tue 06 Nov 2018 06:50:44 PM PST
11: sigtoperl.output       1541559044 Tue 06 Nov 2018 06:50:44 PM PST
12: zerotime               1541559107 Tue 06 Nov 2018 06:51:47 PM PST
13: zerotime.cpp           1541559107 Tue 06 Nov 2018 06:51:47 PM PST
14: zerotime.txt           1541559136 Tue 06 Nov 2018 06:52:16 PM PST
15: .                      1541559136 Tue 06 Nov 2018 06:52:16 PM PST
16: /dev/null              1541003517 Wed 31 Oct 2018 09:31:57 AM PDT
```

```cpp
 1: // $Id: sigtoperl.cpp,v 1.1 2018-11-06 18:50:44-08 - - $
 2:
 3: #include <cstdlib>
 4: #include <cstring>
 5: #include <ctime>
 6: #include <iomanip>
 7: #include <iostream>
 8: #include <string>
 9: #include <sys/utsname.h>
10: using namespace std;
11:
12: int main (int, char** argv) {
13:
14:    // Print UTS system information.
15:    struct utsname utsbuf {};
16:    uname (&utsbuf);
17:    cout << "# " << basename (argv[0]) << ": " << utsbuf.machine
18:         << " " << utsbuf.sysname << " " << utsbuf.nodename << endl;
19:
20:    // Print current date/time.
21:    struct tm tm_local;
22:    time_t tm_now = time (nullptr);
23:    localtime_r (&tm_now, &tm_local);
24:    char tm_buffer[256];
25:    strftime (tm_buffer, sizeof tm_buffer, "%c", &tm_local);
26:    cout << "# " << basename (argv[0]) << ": " << tm_buffer << endl;
27:
28:    // Print strsignal information.
29:    constexpr int MAXSIG = 255;
30:    cout << "my %strsignal = (" << endl;
31:    for (int sig = 0; sig < MAXSIG; ++sig) {
32:       const char* strsig = strsignal (sig);
33:       if (strsig == nullptr) continue;
34:       string stringsig = strsig;
35:       if (stringsig.find_first_of ("Unknown signal ") == 0) continue;
36:       if (stringsig.find_first_of ("Real-time signal ") == 0) continue;
37:       cout << setw(5) << sig << " => \"" << stringsig << "\"," << endl;
38:    }
39:    printf (");\n");
40:    return EXIT_SUCCESS;
41: }
42:
```

```
 1: # sigtoperl: x86_64 Linux unix4.lt.ucsc.edu
 2: # sigtoperl: Tue Nov  6 18:52:17 2018
 3: my %strsignal = (
 4:      1 => "Hangup",
 5:      2 => "Interrupt",
 6:      3 => "Quit",
 7:      4 => "Illegal instruction",
 8:      5 => "Trace/breakpoint trap",
 9:      6 => "Aborted",
10:      7 => "Bus error",
11:      8 => "Floating point exception",
12:      9 => "Killed",
13:     11 => "Segmentation fault",
14:     13 => "Broken pipe",
15:     14 => "Alarm clock",
16:     15 => "Terminated",
17:     16 => "Stack fault",
18:     17 => "Child exited",
19:     18 => "Continued",
20:     19 => "Stopped (signal)",
21:     20 => "Stopped",
22:     21 => "Stopped (tty input)",
23:     22 => "Stopped (tty output)",
24:     24 => "CPU time limit exceeded",
25:     25 => "File size limit exceeded",
26:     26 => "Virtual timer expired",
27:     27 => "Profiling timer expired",
28:     28 => "Window changed",
29:     29 => "I/O possible",
30:     30 => "Power failure",
31:     31 => "Bad system call",
32: );
```