```c

/*
 *
 * Simulation_Run of A Single Server Queueing System
 *
 * Copyright (C) 2014 Terence D. Todd Hamilton, Ontario, CANADA,
 * todd@mcmaster.ca
 *
 * This program is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License as published by the Free
 * Software Foundation; either version 3 of the License, or (at your option)
 * any later version.
 *
 * This program is distributed in the hope that it will be useful, but WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
 * FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License for
 * more details.
 *
 * You should have received a copy of the GNU General Public License along with
 * this program.  If not, see <http://www.gnu.org/licenses/>.
 *
 */

/*****************************************************************************/

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include "output.h"
#include "simparameters.h"
#include "packet_arrival.h"
#include "cleanup_memory.h"
#include "trace.h"
#include "main.h"

/*****************************************************************************/

/*
 * main.c declares and creates a new simulation_run with parameters defined in
 * simparameters.h. The code creates a fifo queue and server for the single
 * server queueuing system. It then loops through the list of random number
 * generator seeds defined in simparameters.h, doing a separate simulation_run
 * run for each. To start a run, it schedules the first packet arrival
 * event. When each run is finished, output is printed on the terminal.
 */

int
main(void)
{
  Simulation_Run_Ptr simulation_run;
  Simulation_Run_Data data;

  /*
   * Declare and initialize our random number generator seeds defined in
```

```
55     * simparameters.h
56     */
57
58    unsigned RANDOM_SEEDS[] = {RANDOM_SEED_LIST, 0};
59    unsigned random_seed;
60    int j=0;
61
62    /*
63     * Loop for each random number generator seed, doing a separate
64     * simulation_run run for each.
65     */
66
67    while ((random_seed = RANDOM_SEEDS[j++]) != 0) {
68
69      simulation_run = simulation_run_new(); /* Create a new simulation run. */
70
71      /*
72       * Set the simulation_run data pointer to our data object.
73       */
74
75      simulation_run_attach_data(simulation_run, (void *) & data);
76
77      /*
78       * Initialize the simulation_run data variables, declared in main.h.
79       */
80
81      data.blip_counter = 0;
82      data.arrival_count = 0;
83      data.number_of_data_packets_processed = 0;
84      data.number_of_voice_packets_processed = 0;
85      data.accumulated_data_packet_delay = 0.0;
86      data.accumulated_voice_packet_delay = 0.0;
87      data.random_seed = random_seed;
88
89      /*
90       * Create the packet buffer and transmission link, declared in main.h.
91       */
92
93      data.data_packet_buffer = fifoqueue_new();
94      data.voice_packet_buffer = fifoqueue_new();
95      data.link   = server_new();
96
97      /*
98       * Set the random number generator seed for this run.
99       */
100
101     random_generator_initialize(random_seed);
102
103     /*
104      * Schedule the initial data and voice packet arrival for the current clock
    time (= 0).
105      */
106
107     schedule_voice_packet_arrival_event(simulation_run,
108       simulation_run_get_time(simulation_run));
```

```
109
110     schedule_packet_arrival_event(simulation_run,
111           simulation_run_get_time(simulation_run));
112
113     /*
114      * Execute events until we are finished.
115      */
116
117     while((data.number_of_data_packets_processed +
    data.number_of_voice_packets_processed) < RUNLENGTH) {
118       simulation_run_execute_event(simulation_run);
119     }
120
121     /*
122      * Output results and clean up after ourselves.
123      */
124
125     output_results(simulation_run);
126     cleanup_memory(simulation_run);
127   }
128
129   getchar();   /* Pause before finishing. */
130   return 0;
131 }
132
133
134
135
136
137
138
139
140
141
142
143
144
```