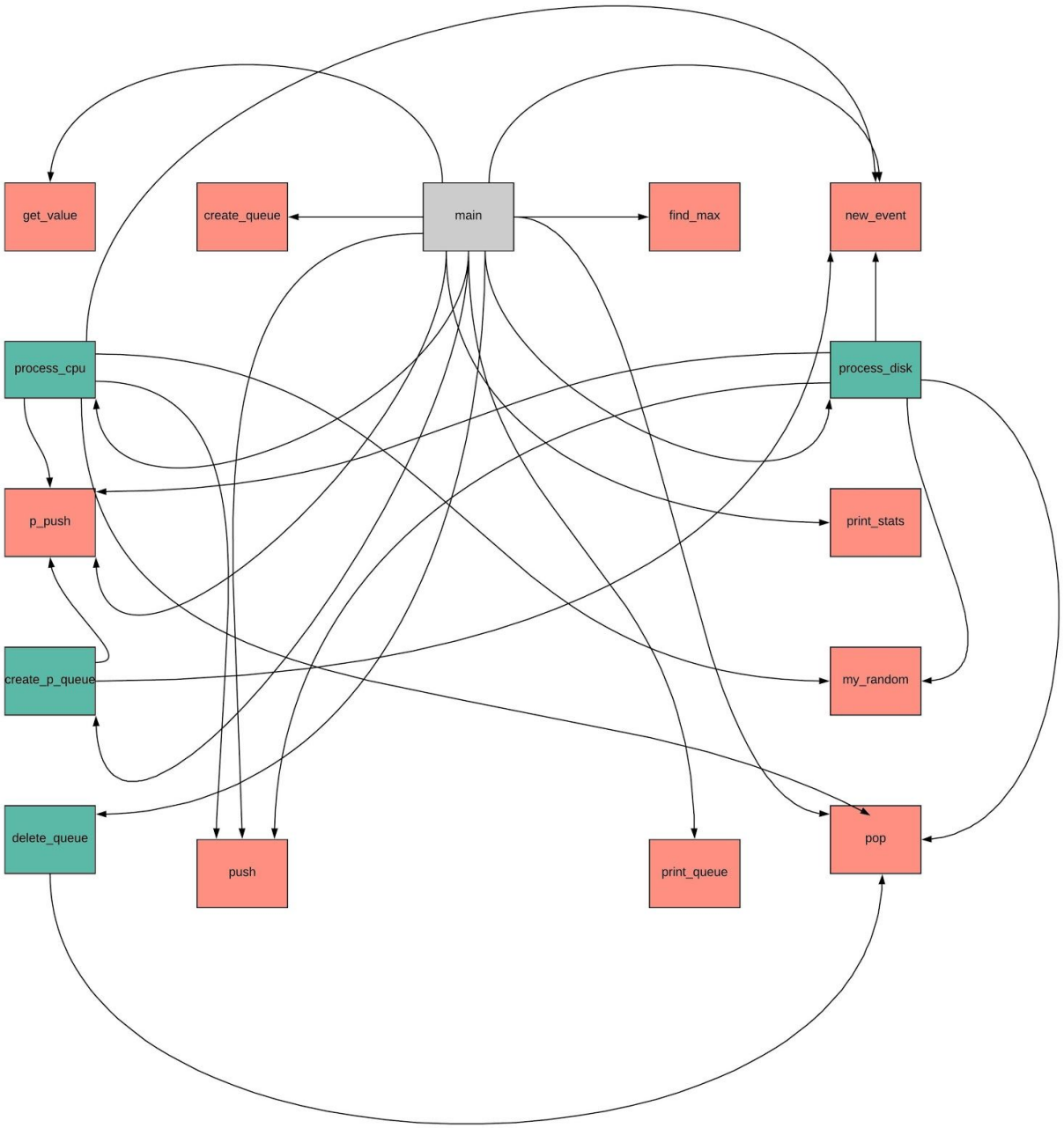
For the data structures in my program, I created four queues. One priority queue, that was prioritized by the time of the events placed into the queue and three FIFI queus. I organized both of these queue types as linked list. I have also, created two structs that would help me to build my queues. The first struct is the event struct. Inside of this struct we have three integers that will represent the event. The first one is the job number. This will help us to follow the job throughout the program. The second is the event type. This will tell us what to do with the job when we pop it off of the priority queue in the program. There are five total event types; ARRIVAL, FINISH, DISK_ARRIVAL, DISK_FINISH, and SIMULATION_END. They are defined as 0,1,2,3,4 respectively. ARRIVAL refers to when the job arrives at the cpu, FINISH refers to when the job completes its work at the cpu, DISK_ARRIVAL refers to when a job arrives at the disk, DISK_FINISH refers to when a job completes it work at one of the disks, and SIMULATION_END refers to when the simulation is complete and we can exit the program. The third integer will represent the time of the job. A number is usually generated at random and added to the current time to get this number. This integer is also how we sort of priority queue. The priority queue is sorted by lowest to highest time. The second struct I created was the queue struct. This contains an event struct, which I just mentioned and a struct pointer to another queue struct. This is where the actual linked list comes into play.

All of the events that occur in the program are recorded into a log file. The events are recorded in the order that they occur. The format of the log entries are as follows:

- At x Job y arrived at the CPU

- At x Job y finished at the CPU.
- At x Job y exited.
- At x Job y arrived at the Disk.
- At x Job y finished at Disk 1.
- At x Job y finished at Disk 2.

Function calls (red = functions are only called, green = functions are called and call other functions, grey = main function)



For testing of the priority queue, I inserted multiple random events in a random order and printed out the queue using a print queue function that I wrote. I then looked through the output to verify that it was in the proper order (increasing time). I also, conducted a similar test to the FIFO queue. I inserted the same random events and printed out the queue. I verified that the queue printed in the same order that I inserted. This can be seen by running the program. To test the random probability generator, I set the quit probability to 20%. Then I set up a few variables to measure how many jobs actually exit. After the program executes I print the quit probability and the experimental quit probability. After doing so, I received a result that was close to 20%. I verified this by changing the quit probability to 50% and received a similar result. This can also be seen by running the program. The statistical results can also be seen by running the program. I have included a sample output below. This data prints out to standard out. This output was gained from the following input.

INPUT:

SEED 5

INIT_TIME 0

FIN_TIME 1000

ARRIVE_MIN 1

ARRIVE_MAX 20

QUIT_PROB 0.2

CPU_MIN 10

CPU_MAX 20

DISK1_MIN 10

DISK1_MAX 30

DISK2_MIN 10

DISK2_MAX 30

OUTPUT:

Priority Queue average 65.677475.

CPU average 797.546204.

Disk 1 average 2.068612.

Disk 2 average 2.446785.

Priority Queue max 99.

CPU max 1652.

Disk 1 max 4.

Disk 2 max 4.

CPU utilization 0.923000.

Disk 1 utilization 0.281000.

Disk 2 utilization 0.260000.

CPU average response time 15.644068.

Disk 1 average response time 12.217391.

Disk 2 average response time 20.000000.

CPU max response time 20.

Disk 1 max response time 30.

Disk 2 max response time 28.

Throughput of CPU 0.059000.

Throughput of Disk 1 0.023000.

Throughput of Disk 2 0.013000.

Quit Probability 0.200000 Actual Quit Probability 0.254237

Priority queue test.

Time 0 Type 0 Job Num 0

Time 23 Type 2 Job Num 15

Time 45 Type 1 Job Num 5

Time 56 Type 3 Job Num 11

Time 500 Type 0 Job Num 9

Time 1000 Type 4 Job Num 0

FIFO queue test.

Time 45 Type 1 Job Num 5

Time 1 Type 2 Job Num 15

Time 56 Type 3 Job Num 11

Time 5 Type 0 Job Num 9