

CS2031 Telecommunications 2

Assignment #1 Command and Control

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Gregory Partridge, 17331009

November 4th, 2019

Contents

1 Introduction. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1

2 Overall Design. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2.1 Command and Control. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

2.2 Broker. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .2

2.3 Worker. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

3 Implementation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

3.1 Task 1. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

3.2 Task 2. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

3.3 Task 3. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

3.4 Task 4. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

3.5 Task 5. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5

3.6 Task 6. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7

4 Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

5 Summary . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

6 Reflection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8

**Introduction**

The task I was given was to create a system in which the Command and control can give assignments to a broker which will then assign them to the workers. When the workers finish their task, they will message the broker that they are done and then the broker will tell the Command and Control. After that the Command and Control can then assign a new task to the broker.

In the following, I will discuss first two mechanisms that are fundamental to my solution. This will be followed by the overall design of the network elements of my solution and a description of the data units that are exchanged between the network elements.

**Overall Design**

The first part I will explain my understanding of command and control, brokers and workers as well as the sample code given. The explanations of these designs will describe their individual elements followed by a description of the packets sent between the terminals.

Command and Control

The command and control are the main form of input and it is where the assignments come in. The command and control will take these inputs and decide what the assignment is and how many workers are necessary for it. It will then convert the assignment string into a packet containing the string length, packet type, source port and finally send the assignment to the broker. It will then wait for confirmation that the broker received the packet.

Broker

The broker acts as a middle man for the worker and the command and control. It will receive messages from both and must differentiate between the two. After this it tries to find the nature of the message and follow the instructions of the message in the case its from the Control and Command. If that is the case it must pass the assignment to the worker or workers.

In the case that it’s from a worker it will determine if its conformation that its work is done, or it is submitting its name to volunteer for work. In the case that it is sending conformation that it is done its assignment, it will check if all the assigned workers are finished their tasks and if so, the broker will message the Control and Command that they are done the task. In the case that it is sending its name it will save the name and the source address of the packet. The broker also records which workers are available to work and which ones are busy.

Worker

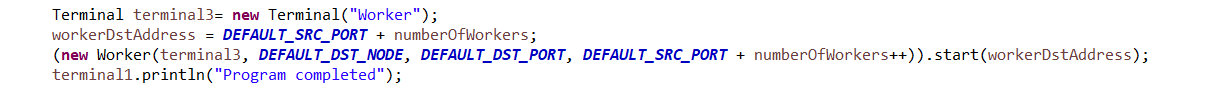
The worker will receive work from the broker and decide if it wants to do the task. If they decide to do the task they will work on the task and when they are finished, they will contact the broker that it has completed the assignment and that is available to work for a new assignment.

**Implementation**

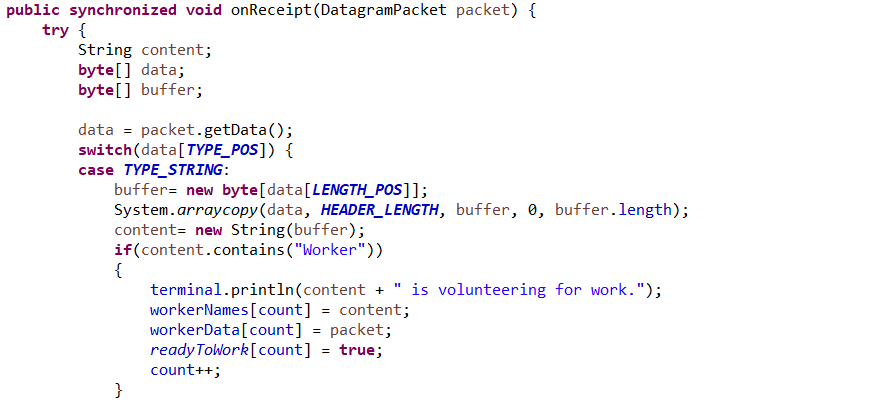
Task 1

1. Workers that accept a name as input, send a message that they are volunteering for work to a broker and print work assignments that have been forwarded by a broker.

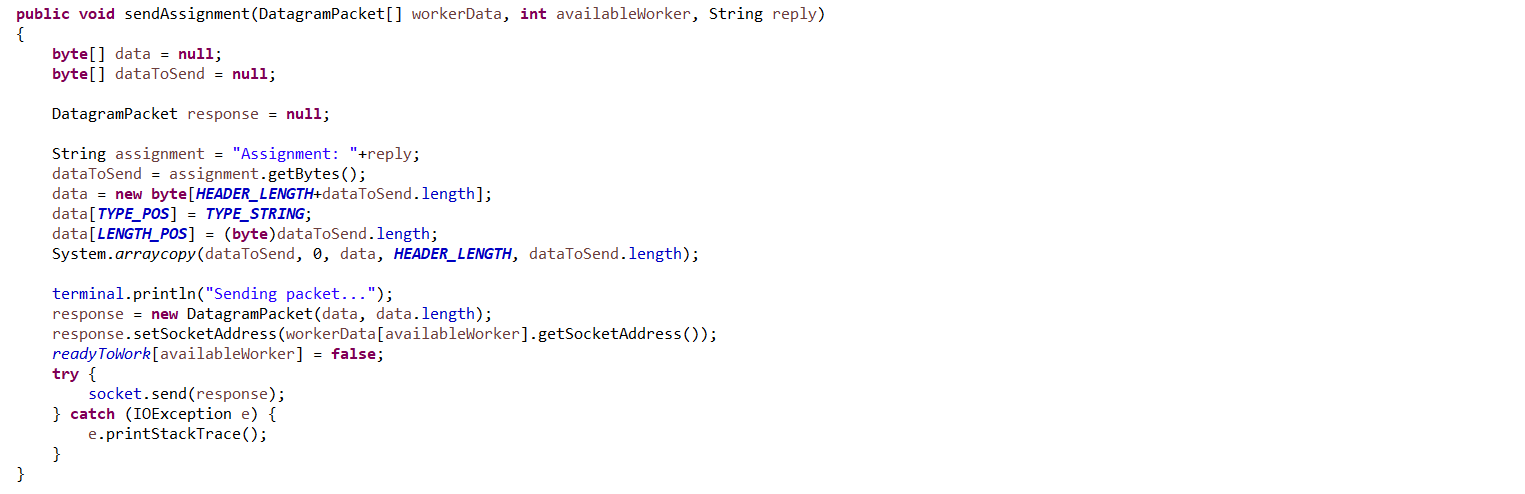
For this assignment I first had to make my worker class. I completed this task by first adding a lot of the content that was available in the Client class in the sample code. I then heavily edited the work and added some extra functions. These functions were onReciet(), sendWorkerName() and initialize the worker. In the main the worker would be initialized with a new terminal for each new worker and each one would contain a unique port. I would do this by keeping track of the number of workers and adding that number on to the default port.



After doing so the new Worker would be asked for its name in the terminal and after that would be asked would it like to volunteer. If they wish to volunteer, I would add a “Worker: “to the start of the string of the worker name and send it to the Broker. Here the Broker would check and see if the String in the data packet contained “Worker: “. If it did I would add the worker name to an array String, the datagram packet to an array of datagram packets and would then set the point in a Boolean array to true as the worker is now available to work.



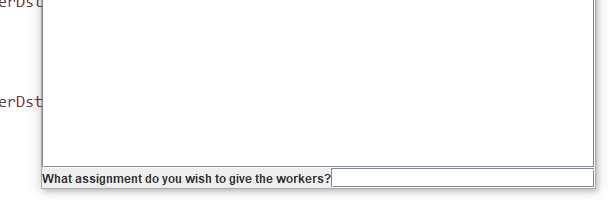
Later in the program the broker will send an assignment to the worker. It does this in the sendAssignment() function. This function requires the workerData array, seeIfWorkerIsBusy() function that just returns the first available worker and a string that contains the assignment. After doing this it enters the function. What it does is send the work assignment to the source port of the first available worker.



Task 2

1. C&C server that accepts a work description as input and transmits a message with the work description to a broker.

The Control and Command is given a list of options which cover for task 4 and you can start the assignment from any of these apart from option 1. By doing so it connects to the broker and the first thing it does is ask for the assignment to be sent out to the workers.



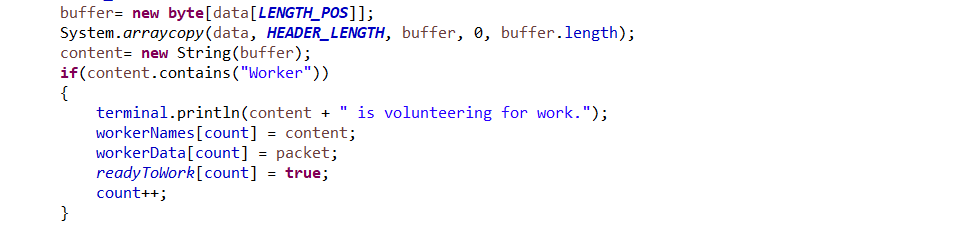


(Actual code for Input)

Task 3

1. A broker that receives messages from workers, maintains lists of available workers and forwards incoming work descriptions to the workers.

When the worker sends the message with their name, they are added to an array containing their original datagram packet which in turn contains the source port of the worker.

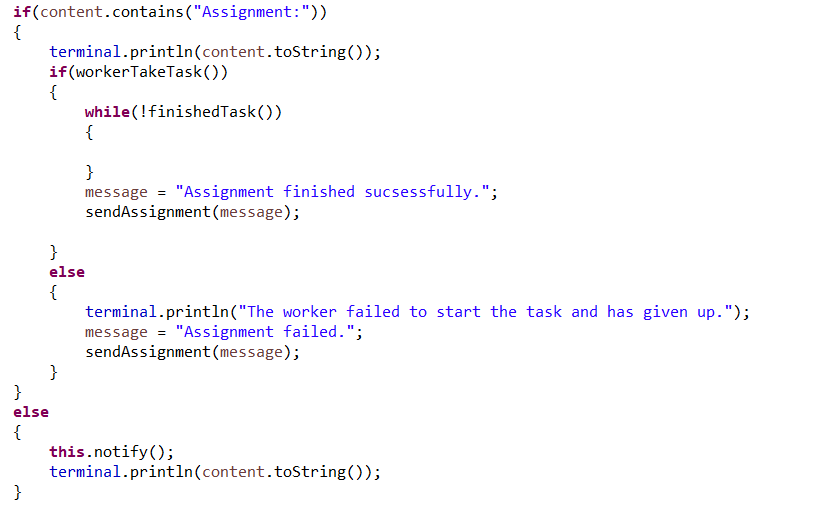


Task 4

1. Workers may choose to reply to the broker with results from work they carried out or withdraw their availability.

My code does not work for this part, but I can still show what I have done to try and get it to work.

The work is assigned to the worker and the worker must choose whether they want to do the assignment. If they choose to do it, they reply when it is done, and the worker sends an assignment to the broker stating if they failed or succeeded the assignment.

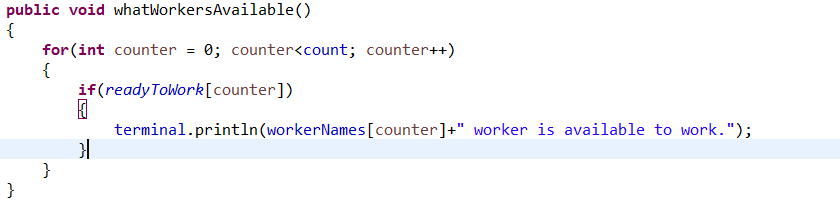


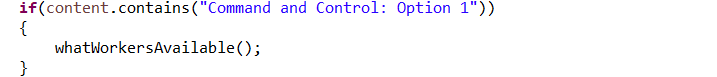
Task 5

1. Work descriptions from a C&C application may request the same work description to be sent to one, a number of workers or all workers

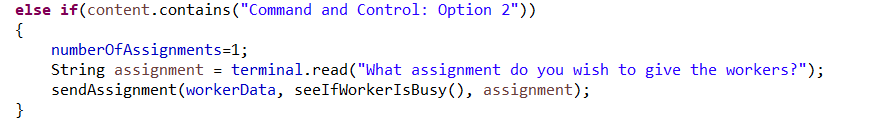
For this I got the input in the Command and Control’s input to choose one of four options. These included check what workers were available, send assignment o one worker, send assignment to all workers and send assignment to set number of workers. The Control and Command would then send strings based on these to the broker and the broker would act dependent on the task.

If it’s the first option, it will call a function that prints all available workers who aren’t working. This is done with a simple for loop that checks the Boolean array to see if it’s available to work.

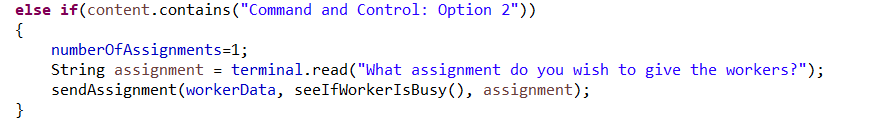




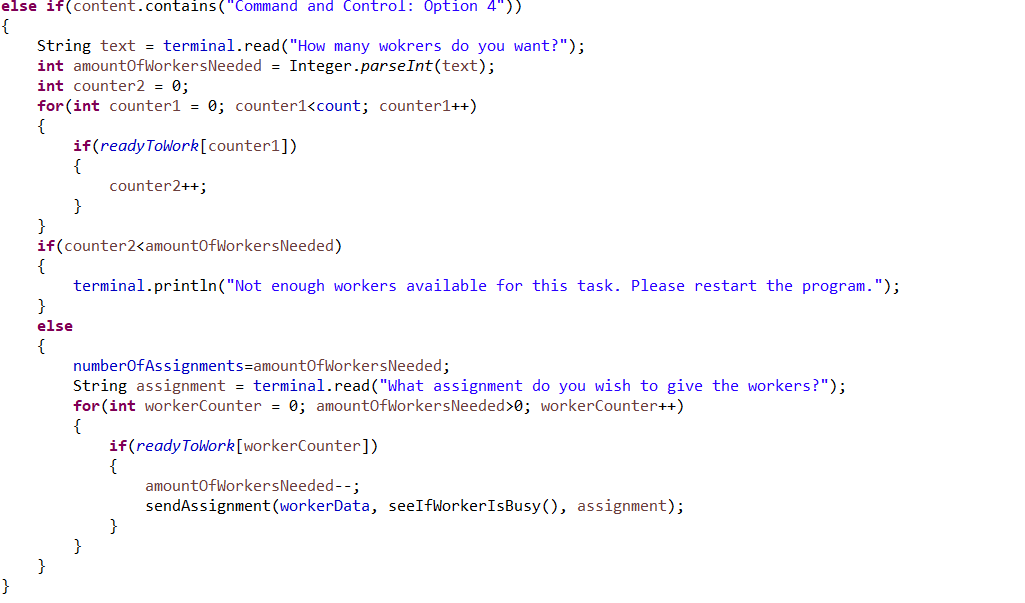
If it is option 2 it will send the assignment to the first available worker.



If option 3 is chosen it will send the assignment to all workers who are available. This is done through a for loop that checks if the worker is working and if so will send them the assignment.



If option 4 is selected it will then request how many workers, you wish for. If it’s over the number of available workers, it will fail. Otherwise it will send the assignments to number of workers asked for.



Task 6

1. The broker and the workers may implement acknowledgements and the C&C may wait for acknowledgements from a broker before proceeding to accept input of another work description.

My code does not work for this part, but I can still show what I have done to try and get it to work.

When the Command and Control sends out the assignment it doesn’t get a response from the broker just yet. The broker then sends out a message to the appropriate worker or workers and will wait for a response from them or else it will not send back the acknowledgment. When all the workers have sent conformation they have completed their tasks the broker will then send an acknowledgment to the Command and Control which will then call the start() function and hence only then allow the Command and Control to send another assignment.

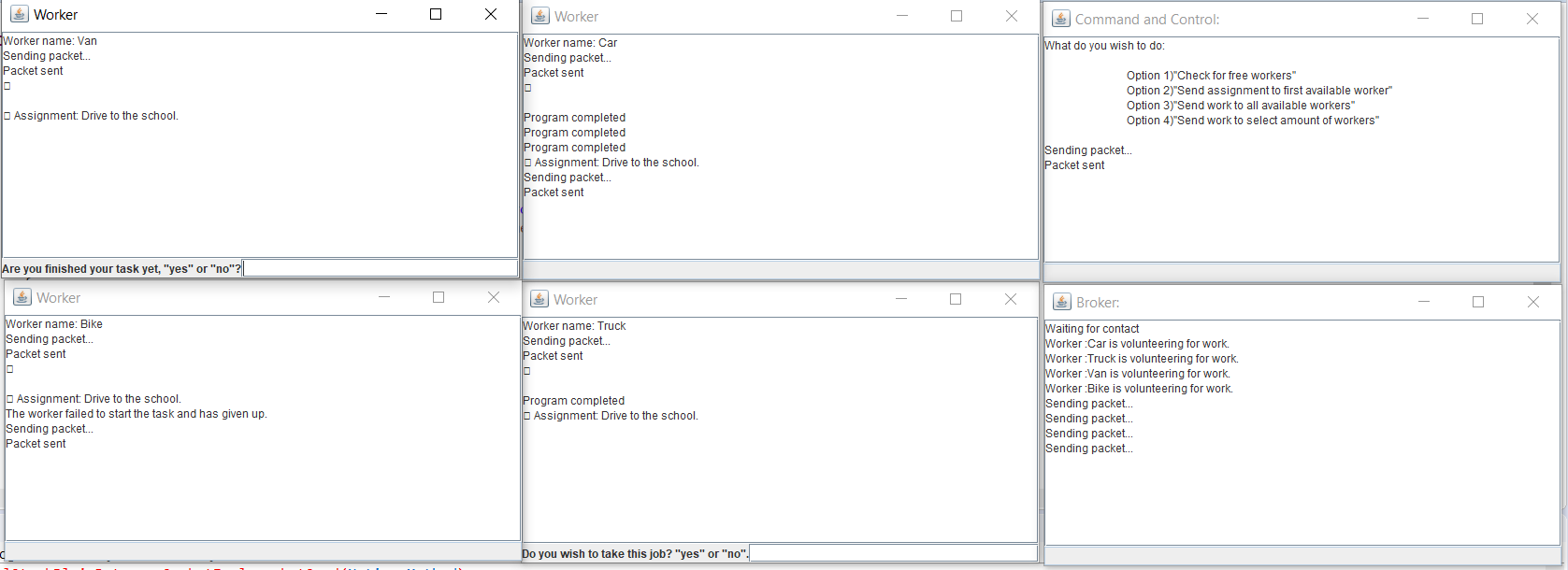


(Command and Control)



(Broker) Requires numberOfAssignmnets==0 for acknowledgment to C&C.

**Discussion**



The strengths of my program are numerous. The fact that the Command and Control does one assignment at a time meaning it’s very clear when an assignment has not been completed to the user. Another strength my program has is that there is an option to see what workers are available from the Control and Command terminal. One other strength of the program is that it can add workers at any time, and they can choose whether they want to take a task or not.

Weaknesses of my project would include that the worker does not send back a confirmation that it completed the assignment correctly after it is done with the task and by extension the Control and Command does not work consistently enough for me to be confident in it. A flaw with the whole wait for assignment to be done before passing more work would be that the program can only run one assignment at a time and hence would not increase in speed as more workers are added. This in turn caps it speed.

**Summary**

The report summarizes how I answered every part of the question I attempted as well as the parts I completed. It shows a viable way to send commands through a client to a server and get workers to complete assignments and afterwards give conformation and complete a new task. The description of the implementation in this document highlights the essential components of my solution.

**Reflection**

I found my project was expectable but if I were to do it again, I would have started earlier as well as try and have the base of my code to be tidier and clear. Looking back though I really liked the lack of a time limitation on the project except for the final deadline. I could really base my work schedule around me, and the assignment was a lot more relaxed. After doing the assignment I definitely feel I have a better understanding on this area of telecommunication and am definitely confident going into my next assignment that I can improve on my work.