COMP9331 Assignment Final Report
z3489044

This is an extension version . There are seven classes in total
The overall idea of this program is that.
Step1:
Creating test .sh file, creating nodes with identifiers and its first successor and second successor . Creating links of them
Step2 :
Sending ping request to its first successor and its second successor every 5 seconds . After receiving a ping request (which has a parameter to indicate this request comes from its first pre-successor or the second pre-successor).Then setting its first pre-successor and second pre-successor with this two parameters . Using the method in Node set_fir_pre_suc() and set_sec_pre_suc()

Step3:
Checking current node by function check_function.
If it have the file required , then send a TCP packet to the source_port ,which is the first node that requests the file.
else, sending request TCP packets to its first and second successor to check whether they have until found the file
Step4
If the user types quit
Current node will send a TCP quit packet to its first pre-successor and second pre-successor. This TCP quit packet contains Quit_COMMEND, the id_of_fir_successor, the id_of_sec_successor,current_node_id. Then its fir_pre successor and second pre_successor will receive the Quit_COMMEND, and response to current node that they have already updated the information of their successors .Then after making sure that both its two pre-successors have updated the information , current system will call the function System.exit(0);
Step5:
Each node after sending a ping_request to its successors , its two parameters ,which are force_quit_condition will + 1(which is 0 initially). and After receiving a ping_response , this parameter will be settled to 0 (Because this means that its successor is still alive ).
But if the condition parameter larger than 3 ,which means its successor is very possible to be lost , then it will check which successor is lost Because the two different successor have different parameters

They are force_quit_conditon1 and force_quit_condition2, represents first successor and second successor respectively .And then if first successor is lost , then it will ask its sec_successors first successor as it second successor , and let its original second successor as its first successor .
If the second successor if lost ,
it will ask its fir successor's second successor to be his second successor

This is the detail classes of my project.
1Class Node:
This class is basic nodes of the linked-list-like structure .
It has attributes int id
Node fir – suc , sec_suc, fir_pre, and sec_pre.
2Class Timer
This is a sub class of thread , this is used to calculating the time .
it will wake the udpsender class every 5 seconds to let the udpsender send ping request to its two successor .
3UDPsender
This class is especially used to send UDP packets , this is a subclass of Thread class, it will keep suspending unless there is another object resume it . After it was called , it will check the message queue to make sure which action to take.
Such as attribute PING_REQUEST is to send ping request to its two successor .
4UDPreceiver
This is also a subclass of Thread , it keeps listening to the UDP DatagramSocket analyzing the commend it received . Such as if the parameter is ping_request , then it will call UDPsender to send a response to the source_port
5TCP sender
Similar to UDPsender , it also keeps suspending ,until there are some other process call it . and sending TCPpackets according to the commend parameter in the message queue .
6TCPreceiver
It has a judge function check_function(int filename)
It returns true if the file is in the node , else
return false

It keeps listening to the TCP socket , and analyzing the commend which is received in the socket .
Such as requesting of file,if this file is not in the node , then it will call TCP sender to send requesting packet to its successors .
7cdht_ex
This is the main class , it has plenty of condition parameters to indicate the UDPsender, UDPreceiver, TCP sender and TCP receiver what to do .
And it stars the 5 process , UDPsender, UDPreceiver, TCPsender,TCPreceiver,Timer.
And keeps listenting to the user input.