

Computer Networks Lab0

黄嘉祺 221220108

1 Program Structure and Design

1.1 webget

- 我在阅读了 `util/socket.hh` , `util/file_descriptor.hh` 和 `util/address.hh` 中定义的 `FileDescriptor` , `Socket` , `TCPSocket` 和 `Address` 等类和它们提供的函数接口代码后, 发现要实现 `get_URL` 函数, 需要使用 `Address` 的构造函数来解析主机地址, 随后使用 `TCPSocket` 提供的连接函数来连接主机, 最后使用 `Socket` 提供的读写函数来像主机发送请求并读出收到的回复
- 在读取回复时要注意检测数据流是否结束 (即是否读到 `eof`) , 因此使用 `socket.eof()` 函数来检测
- 基于以上思路我补充实现了 `apps/webget.cc` 中的 `get_URL` 函数, 实现了从URL中解析出主机地址, 连接主机, 发送请求并读取回复的功能
-

1.2 An in-memory reliable byte stream

- 阅读 `src/byte_stream.cc` 和 `src/byte_stream.hh` 的源码后发现试验任务是需要实现 `ByteStream` 类及其派生的 `Writer` 和 `Reader` 类已经定义好的一些成员函数接口, 以完整实现这个类
- 发现在实现这些函数的时候需要给 `byteStream` 类再添加一些成员变量用于存储和表达这些函数的功能, 因此添加了 `num_bytes_` (数据流中当前字节数), `num_bytes_pushed_` (被推送进数据流的字节数), `num_bytes_popped_` (被弹出数据流的字节数), `close_flag_` (数据流是否关闭) 等成员变量, 并添加了一个 `std::string` 类型的 `bytes_` 变量用于存储数据流中的数据内容
- 函数的实现主要是对这些成员变量的读取或修改等操作, 例如 `push` 函数会将数据写入 `bytes_` 中并更新 `num_bytes_` 和 `num_bytes_pushed_` 等成员变量, `pop` 函数会将数据从 `bytes_` 中删除并更新 `num_bytes_` 和 `num_bytes_popped_` 等成员变量等
- 基于以上思路我实现了 `src/byte_stream.cc` 中的 `ByteStream` 类及其派生的 `Writer` 和 `Reader` 类的成员函数接口, 实现了一个在内存中的可靠字节流

2 Implementation Challenges

- 在完成 'An in-memory reliable byte stream' 部分的时候, 一开始我将新定义成员变量的初始化放在了构造函数的初始化列表中进行初始化, 然而这样做会导致编译错误, 因为这些成员变量的初始化需要依赖于其他成员变量的初始化, 而这些成员变量的初始化顺序是按照它们在类中的声明顺序进行的, 因此我将这些成员变量的初始化放在了类的定义时, 解决了这个问题

3 Remaining Bugs

目前我的代码已经通过了全部的测试，暂时未发现明显的bug

4 Experimental Results and Performance

- Fetch a Web page

```
connection closed by foreign host.
lefty777@EDVAC-2023:~$ telnet cs144.keithw.org http
Trying 104.196.238.229...
Connected to cs144.keithw.org.
Escape character is '^]'.
GET /hello HTTP/1.1
Host: cs144.keithw.org
Connection: close

HTTP/1.1 200 OK
Date: Mon, 23 Sep 2024 11:04:54 GMT
Server: Apache
Last-Modified: Thu, 13 Dec 2018 15:45:29 GMT
ETag: "e-57ce93446cb64"
Accept-Ranges: bytes
Content-Length: 14
Connection: close
Content-Type: text/plain

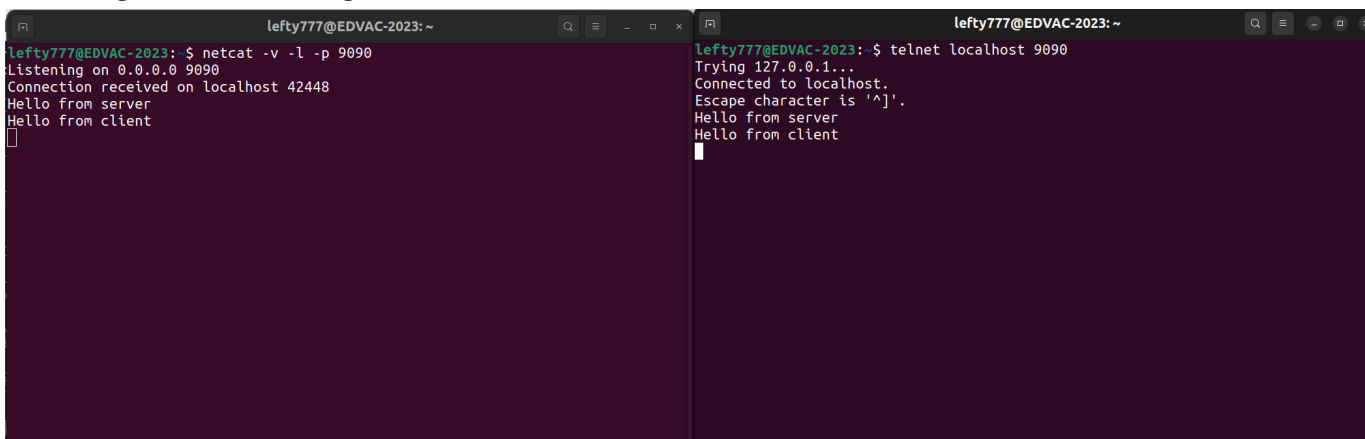
Hello, CS144!
Connection closed by foreign host.
lefty777@EDVAC-2023:~$
```

- Send yourself an email

```
rcpt to:<2788990293@qq.com>
250 OK
DATA
354 End data with <CR><LF>.<CR><LF>.
from:2788990293@qq.com
to:2788990293@qq.com
subject: lab0 test telnet and smtp

hello from cs144
.
250 OK: queued as.
```

- Listening and connecting



```
lefty777@EDVAC-2023:~$ netcat -v -l -p 9090
Listening on 0.0.0.0 9090
Connection received on localhost 42448
Hello from server
Hello from client

lefty777@EDVAC-2023:~$ telnet localhost 9090
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Hello from server
Hello from client
```

- Writing webget: Passed all the tests

```
file_descriptor.hh  webget.cc x  byte_stream.hh  byte_stream.cc  byte_stream_basics.cc  socket.hh  socket.cc  address.hh
apps > webget.cc > get_URL(const string &, const string &)
10 void get_URL( const string& host, const string& path )
29 // cerr << "EOF reached\n";
30 // } else {
31 // socket.read( data );
32 // cerr << "Data read: " << data << "\n";
33 // }
34 socket.write( "Connection: close\r\n" );
35 // if ( socket.eof() ) {
36 // cerr << "EOF reached\n";
37 // } else {
38 // socket.read( data );
39 // cerr << "Data read: " << data << "\n";
40 // }
41 socket.write( "\r\n" );
42 while ( !socket.eof() ) {
43     socket.read( data );
44     cout << data;
45 }
46 socket.close();
47 // cerr << "Function called: get_URL(" << host << ", " << path << ")\n";
48 // cerr << "Warning: get_URL() has not been implemented yet.\n";
49
50
问题 输出 调试控制台 终端 端口 注释
bash-build + -  0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500  501  502  503  504  505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546  547  548  549  550  551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  587  588  589  590  591  592  593  594  595  596  597  598  599  600  601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617  618  619  620  621  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649  650  651  652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688  689  690  691  692  693  694  695  696  697  698  699  700  701  702  703  704  705  706  707  708  709  710  711  712  713  714  715  716  717  718  719  720  721  722  723  724  725  726  727  728  729  730  731  732  733  734  735  736  737  738  739  740  741  742  743  744  745  746  747  748  749  750  751  752  753  754  755  756  757  758  759  760  761  762  763  764  765  766  767  768  769  770  771  772  773  774  775  776  777  778  779  780  781  782  783  784  785  786  787  788  789  790  791  792  793  794  795  796  797  798  799  800  801  802  803  804  805  806  807  808  809  810  811  812  813  814  815  816  817  818  819  820  821  822  823  824  825  826  827  828  829  830  831  832  833  834  835  836  837  838  839  840  841  842  843  844  845  846  847  848  849  850  851  852  853  854  855  856  857  858  859  860  861  862  863  864  865  866  867  868  869  870  871  872  873  874  875  876  877  878  879  880  881  882  883  884  885  886  887  888  889  890  891  892  893  894  895  896  897  898  899  900  901  902  903  904  905  906  907  908  909  910  911  912  913  914  915  916  917  918  919  920  921  922  923  924  925  926  927  928  929  930  931  932  933  934  935  936  937  938  939  940  941  942  943  944  945  946  947  948  949  950  951  952  953  954  955  956  957  958  959  960  961  962  963  964  965  966  967  968  969  970  971  972  973  974  975  976  977  978  979  980  981  982  983  984  985  986  987  988  989  990  991  992  993  994  995  996  997  998  999  1000

Lefty777@EDVAC-2023:~/minnow$ cd build/
Lefty777@EDVAC-2023:~/minnow/build$ ./apps/webget cs144.keithw.org /hello
HTTP/1.1 200 OK
Date: Mon, 23 Sep 2024 11:11:16 GMT
Server: Apache
Last-Modified: Thu, 13 Dec 2018 15:45:29 GMT
ETag: "e-57ce93446cb64"
Accept-Ranges: bytes
Content-Length: 14
Connection: close
Content-Type: text/plain

Hello, CS144!
Lefty777@EDVAC-2023:~/minnow/build$
```

```
file_descriptor.hh  webget.cc x  byte_stream.hh  byte_stream.cc  byte_stream_basics.cc  socket.hh  socket.cc  address.hh
apps > webget.cc > get_URL(const string &, const string &)
10 void get_URL( const string& host, const string& path )
29 // cerr << "EOF reached\n";
30 // } else {
31 // socket.read( data );
32 // cerr << "Data read: " << data << "\n";
33 // }
34 socket.write( "Connection: close\r\n" );
35 // if ( socket.eof() ) {
36 // cerr << "EOF reached\n";
37 // } else {
38 // socket.read( data );
39 // cerr << "Data read: " << data << "\n";
40 // }
41 socket.write( "\r\n" );
42 while ( !socket.eof() ) {
43     socket.read( data );
44     cout << data;
45 }
46 socket.close();
47 // cerr << "Function called: get_URL(" << host << ", " << path << ")\n";
48 // cerr << "Warning: get_URL() has not been implemented yet.\n";
49
50
问题 输出 调试控制台 终端 端口 注释
bash + -  0  1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491  492  493  494  495  496  497  498  499  500  501  502  503  504  505  506  507  508  509  510  511  512  513  514  515  516  517  518  519  520  521  522  523  524  525  526  527  528  529  530  531  532  533  534  535  536  537  538  539  540  541  542  543  544  545  546  547  548  549  550  551  552  553  554  555  556  557  558  559  560  561  562  563  564  565  566  567  568  569  570  571  572  573  574  575  576  577  578  579  580  581  582  583  584  585  586  587  588  589  590  591  592  593  594  595  596  597  598  599  600  601  602  603  604  605  606  607  608  609  610  611  612  613  614  615  616  617  618  619  620  621  622  623  624  625  626  627  628  629  630  631  632  633  634  635  636  637  638  639  640  641  642  643  644  645  646  647  648  649  650  651  652  653  654  655  656  657  658  659  660  661  662  663  664  665  666  667  668  669  670  671  672  673  674  675  676  677  678  679  680  681  682  683  684  685  686  687  688  689  690  691  692  693  694  695  696  697  698  699  700  701  702  703  704  705  706  707  708  709  710  711  712  713  714  715  716  717  718  719  720  721  722  723  724  725  726  727  728  729  730  731  732  733  734  735  736  737  738  739  740  741  742  743  744  745  746  747  748  749  750  751  752  753  754  755  756  757  758  759  760  761  762  763  764  765  766  767  768  769  770  771  772  773  774  775  776  777  778  779  780  781  782  783  784  785  786  787  788  789  790  791  792  793  794  795  796  797  798  799  800  801  802  803  804  805  806  807  808  809  810  811  812  813  814  815  816  817  818  819  820  821  822  823  824  825  826  827  828  829  830  831  832  833  834  835  836  837  838  839  840  841  842  843  844  845  846  847  848  849  850  851  852  853  854  855  856  857  858  859  860  861  862  863  864  865  866  867  868  869  870  871  872  873  874  875  876  877  878  879  880  881  882  883  884  885  886  887  888  889  890  891  892  893  894  895  896  897  898  899  900  901  902  903  904  905  906  907  908  909  910  911  912  913  914  915  916  917  918  919  920  921  922  923  924  925  926  927  928  929  930  931  932  933  934  935  936  937  938  939  940  941  942  943  944  945  946  947  948  949  950  951  952  953  954  955  956  957  958  959  960  961  962  963  964  965  966  967  968  969  970  971  972  973  974  975  976  977  978  979  980  981  982  983  984  985  986  987  988  989  990  991  992  993  994  995  996  997  998  999  1000

Lefty777@EDVAC-2023:~/minnow/build$ cd ..
Lefty777@EDVAC-2023:~/minnow$ cmake --build build --target check_webget
Test project /home/Lefty777/minnow/build
Start 1: compile with bug-checkers
1/2 Test #1: compile with bug-checkers ..... Passed    0.25 sec
Start 2: t_webget
2/2 Test #2: t_webget ..... Passed    1.30 sec

100% tests passed, 0 tests failed out of 2

Total Test time (real) = 1.56 sec
Built target check_webget
Lefty777@EDVAC-2023:~/minnow$
```

- An in-memory reliable byte stream: Passed all the tests

file_descriptor.hhwebget.cc ×byte_stream.hhbyte_stream.ccbyte_stream_basics.csocket.hsocket.caddress.h

apps > webget.cc > get_URL(const string &, const string &)
10 void get_URL(const string& host, const string& path)
29 // cerr << "EOF reached\n";
30 // } else {
31 // socket.read(data);
32 // cerr << "Data read: " << data << "\n";
33 // }
34 socket.write("Connection: close\r\n");
35 // if (socket.eof()) {
36 // cerr << "EOF reached\n";

问题输出调试控制台终端端口注释
bash + - x
lefty777@EDVAC-2023:~/minnow\$ cmake --build build --target check0
Test project /home/lefty777/minnow/build
Start 1: compile with bug-checkers
1/10 Test #1: compile with bug-checkers Passed 0.21 sec
Start 2: t webget
2/10 Test #2: t webget Passed 1.26 sec
Start 3: byte_stream_basics
3/10 Test #3: byte_stream_basics Passed 0.01 sec
Start 4: byte_stream_capacity
4/10 Test #4: byte_stream_capacity Passed 0.01 sec
Start 5: byte_stream_one_write
5/10 Test #5: byte_stream_one_write Passed 0.02 sec
Start 6: byte_stream_two_writes
6/10 Test #6: byte_stream_two_writes Passed 0.02 sec
Start 7: byte_stream_many_writes
7/10 Test #7: byte_stream_many_writes Passed 0.06 sec
Start 8: byte_stream_stress_test
8/10 Test #8: byte_stream_stress_test Passed 0.03 sec
Start 37: compile with optimization
9/10 Test #37: compile with optimization Passed 0.09 sec
Start 38: byte_stream_speed_test
ByteStream throughput: 2.74 Gbit/s
10/10 Test #38: byte_stream_speed_test Passed 0.10 sec

100% tests passed, 0 tests failed out of 10

Total Test time (real) = 1.81 sec
Built target check0
lefty777@EDVAC-2023:~/minnow\$

没有通知