

```
1: all: ps7a
2:
3: ps7a: main.o InTouchDeviceStartUp.o
4:         g++ -o ps7a InTouchDeviceStartUp.o main.o -lboost_regex -lboost_date_time
5:
6: main.o: main.cpp InTouchDeviceStartUp.hpp
7:         g++ -c main.cpp -Wall -ansi -pedantic
8:
9: InTouchDeviceStartUp.o: InTouchDeviceStartUp.cpp InTouchDeviceStartUp.hpp
10:        g++ -c InTouchDeviceStartUp.cpp -Wall -ansi -pedantic
11:
12: clean:
13:        rm *.o ps7a
```

```
1: //
2: //  main.cpp
3: //  ps7a
4: //
5: //  Created by Jingxian Shi on 4/12/18.
6: //  Copyright © 2018 Jingxian Shi. All rights reserved.
7: //
8: #include <boost/regex.hpp>
9: #include <iostream>
10: #include <string>
11: #include <vector>
12: #include <fstream>
13: #include "InTouchDeviceStartUp.hpp"
14:
15: int main(int argc, char* argv[]) {
16:     if (argc != 2) {
17:         std::cerr << "Wrong number of arguments arguments." << std::endl;
18:         return -1;
19:     }
20:     std::ifstream fin;
21:     fin.open(argv[1]);
22:     if (fin.fail()) {
23:         std::cerr << "Fail to open file: " << argv[1] << std::endl;
24:         return -1;
25:     }
26:     std::string output_name(argv[1]);
27:     output_name += ".rpt";
28:     std::ofstream fout;
29:     fout.open(output_name.c_str());
30:     if (fout.fail()) {
31:         std::cerr << "Fail to output file: " << output_name << std::endl;
32:         return -1;
33:     }
34:     int lines_scanned = 0;
35:     int boot_started = 0;
36:     int boot_completed = 0;
37:     std::string line;
38:     std::vector<InTouch> boots;
39:     while (std::getline(fin, line)) {
40:         lines_scanned++;
41:         if (regex_match(line, startRegex)) {
42:             boot_started++;
43:             InTouch its(line, argv[1], lines_scanned);
44:             boots.push_back(its);
45:         } else if (regex_match(line, completedRegex)) {
46:             boot_completed++;
47:             boots.back().completed(line, lines_scanned);
48:         }
49:     }
50:     for (int i = 0; i < boots.size() - 1; ++i) {
51:         fout << boots[i] << std::endl;
52:     }
53:     fout << boots[boots.size() - 1];
54:     fin.close();
55:     fout.close();
56:
57:     return 0;
58: }
```

```
1: //
2: // InTouchDeviceStartUp.cpp
3: // ps7a
4: //
5: // Created by Jingxian Shi on 4/12/18.
6: // Copyright © 2018 Jingxian Shi. All rights reserved.
7: //
8: #include <boost/date_time/posix_time/posix_time.hpp>
9: #include <iostream>
10: #include <string>
11: #include "InTouchDeviceStartUp.hpp"
12:
13: InTouch::InTouch(std::string start_line, std::string file_name, int line) {
14:     _log_file_name = file_name;
15:     _start_line = line;
16:     _end_line = 0;
17:     _end_time = "";
18:     _boot_time = 0;
19:     _completed = false;
20:     boost::smatch sm;
21:     boost::regex_match(start_line, sm, startRegex);
22:     _start_time = sm[1];
23: }
24: void InTouch::completed(std::string completed, int line) {
25:     _completed = true;
26:     _end_line = line;
27:     boost::smatch sm;
28:     boost::regex_match(completed, sm, completedRegex);
29:     _end_time = sm[1];
30:     _boot_time = elapsedTime();
31: }
32: int InTouch::elapsedTime() {
33:     boost::posix_time::ptime start_time
34:         (boost::posix_time::time_from_string(_start_time));
35:     boost::posix_time::ptime end_time
36:         (boost::posix_time::time_from_string(_end_time));
37:     boost::posix_time::time_duration elapsedTime = end_time - start_time;
38:     return elapsedTime.total_milliseconds();
39: }
40: std::ostream& operator<<(std::ostream& out, const InTouch& right) {
41:     out << "=== Device boot ===" << std::endl;
42:     out << right._start_line << "(" << right._log_file_name << "): "
43:     << right._start_time << " Boot Start" << std::endl;
44:     if (right._completed) {
45:         out << right._end_line << "(" << right._log_file_name << "): "
46:         << right._end_time << " Boot Completed" << std::endl
47:         << "\tBoot Time: " << right._boot_time << "ms " << std::endl;
48:     } else {
49:         out << "***** Incomplete boot *****" << std::endl;
50:     }
51:     return out;
52: }
```

```
1: //
2: // InTouchDeviceStartUp.hpp
3: // ps7a
4: //
5: // Created by Jingxian Shi on 4/12/18.
6: // Copyright © 2018 Jingxian Shi. All rights reserved.
7: //
8:
9: #ifndef InTouchDeviceStartUp_hpp
10: #define InTouchDeviceStartUp_hpp
11:
12: #include <boost/regex.hpp>
13: #include <stdio.h>
14: #include <string>
15:
16: const std::string start =
17: "(\d{4}\-(\d{2})\-(\d{2}) "
18: "(\d{2}):(\d{2}):(\d{2})): "
19: "\\(log.c.166\\) server started\\s*";
20: const std::string completed =
21: "(\d{4}\-(\d{2})\-(\d{2}) "
22: "(\d{2}):(\d{2}):(\d{2}))\\.(\d{3})"
23: ".*oejs\\.AbstractConnector:Started\\s+"
24: "SelectChannelConnector@0.0.0.0:9080\\s*";
25: const boost::regex startRegex(start);
26: const boost::regex completedRegex(completed);
27:
28: class InTouch {
29: public:
30:     InTouch(std::string start_line, std::string file_name, int line);
31:     void completed(std::string completed, int line);
32:     int elapsedTime();
33:     friend std::ostream& operator<<(std::ostream& out, const InTouch& right);
34: private:
35:     int _start_line;
36:     int _end_line;
37:     int _boot_time;
38:     std::string _log_file_name;
39:     std::string _start_time;
40:     std::string _end_time;
41:     bool _completed;
42: };
43:
44: #endif /* InTouchDeviceStartUp_hpp */
```