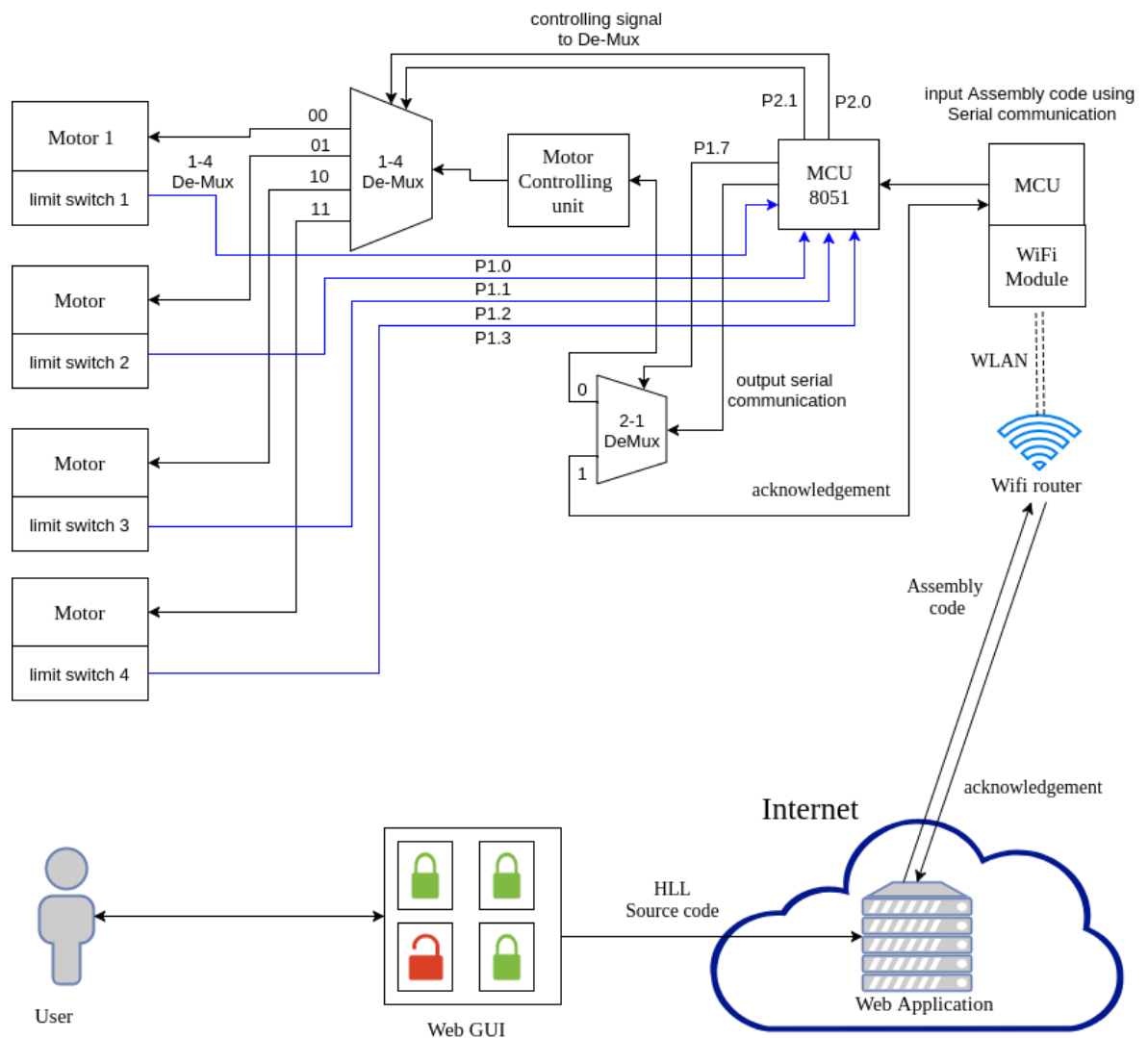


UML diagram of IOT application, which can control door locking system by web interface



All the door locks are connected to servo motors, so these motors can be control by the MCU. This MCU is have a Wi-Fi module which can sync with the web application through internet.

Production rules as follows,

S --> door<NT1> ; | CHECK <NT2> ;

NT1 --> <NT2> <NT3>

NT2 --> [0-3]

NT3 --> lock | unlock

Let's indicate,

check--> r
door--> d
lock--> l
unlock--> u
; --> s

Grammar (G) = {N, T, P, S}

N = set of non terminal

T = set of terminal

P = set of production rules

S = Starting symbol

N = {S, NT1, NT2, NT3}

T = {d, r, l, u, [0-3], s}

P = { S --> d<NT1> s | r<NT2> s,
NT1 --> <NT2> <NT3> ,
NT2 --> [1-4]
NT3 --> l | u }

header file

```
typedef enum { typeCon, typeId, typeOpr } nodeEnum;

/* constants */
typedef struct {
    int value; /* value of constant */
} conNodeType;

/* identifiers */
typedef struct {
    int i; /* subscript to sym array */
} idNodeType;

/* operators */
typedef struct {
    int oper; /* operator */
    int nops; /* number of operands */
    struct nodeTypeTag *op[1]; /* operands, extended at runtime */
} oprNodeType;

typedef struct nodeTypeTag {
    nodeEnum type; /* type of node */

    union {
        conNodeType con; /* constants */
        idNodeType id; /* identifiers */
        oprNodeType opr; /* operators */
    };
} nodeType;

extern int sym[26];
```

Bison file

```
%{
/* headerfiles */
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include "doorlock.h"

/* prototypes */
nodeType *opr(int oper, int nops, ...);
nodeType *id(int i);
nodeType *con(int value);
void freeNode(nodeType *p);
int ex(nodeType *p);
int yylex(void);

void yyerror(char *s);
int sym[26];          /* symbol table */
%}

%union {
    int iValue;          /* integer value */
    char sIndex;         /* symbol table index */
    nodeType *nPtr;      /* node pointer */
};

%token <iValue> NO
%token <sIndex> STATE
%token DOOR CHECK
%left LOCK UNLOCK

%type <nPtr> stmt door_no state

%%

program:      function                { exit(0); };

function: function stmt                { ex($2); freeNode($2); }
        | /* NULL */
        ;

stmt:DOOR door_no state ';'            { $$ = opr(DOOR, 2, $2, $3); }
      | CHECK door_no ';'              { $$ = opr(CHECK, 1, $2); }
      ;

door_no:      NO                      { $$ = con($1); };

state:        LOCK                    { $$ = id(1); }
            | UNLOCK                  { $$ = id(2); }
            ;

%%

nodeType *con(int value) {
    nodeType *p;

    /* allocate node */
    if ((p = malloc(sizeof(nodeType))) == NULL)
        yyerror("out of memory");

    /* copy information */
    p->type = typeCon;
    p->con.value = value;

    return p;
}
```

```

}

nodeType *id(int i) {
    nodeType *p;

    /* allocate node */
    if ((p = malloc(sizeof(nodeType))) == NULL)
        yyerror("out of memory");

    /* copy information */
    p->type = typeId;
    p->id.i = i;

    return p;
}

nodeType *opr(int oper, int nops, ...) {
    va_list ap;
    nodeType *p;
    int i;

    /* allocate node, extending op array */
    if ((p = malloc(sizeof(nodeType) + (nops-1) * sizeof(nodeType *))) == NULL)
        yyerror("out of memory");

    /* copy information */
    p->type = typeOpr;
    p->opr.oper = oper;
    p->opr.nops = nops;
    va_start(ap, nops);
    for (i = 0; i < nops; i++)
        p->opr.op[i] = va_arg(ap, nodeType*);
    va_end(ap);
    return p;
}

void freeNode(nodeType *p) {
    int i;

    if (!p) return;
    if (p->type == typeOpr) {
        for (i = 0; i < p->opr.nops; i++)
            freeNode(p->opr.op[i]);
    }
    free(p);
}

void yyerror(char *s) {
    fprintf(stdout, "%s\n", s);
}

int main(void) {
    yyparse();
    return 0;
}

```

Flex File

```

%{
#include <stdlib.h>
#include "doorlock.h"
#include "y.tab.h"
void yyerror(char *);
%}

```



```

        case CHECK: printf("\nDATA P1 090H\n");
                    printf("ORG 30H\n\n");
                    printf("\tSETB P1.7\n");
                    printf("\tJB P1.");
                    ex(p->opr.op[0]);
                    printf(", LOCK\n\n");
                    printf("UNLOCK: MOV DPTR, #100H\n\n");
                    printf("\tMOV R1, #6\n");
                    printf("\tSJMP LOOP\n");
                    printf("LOCK:  MOV DPTR, #103H\n\n");
                    printf("\tMOV R1, #4\n");
                    printf("\tSJMP LOOP\n");
                    printf("LOOP:  CLR A\n");
                    printf("\tMOVC A,@A+DPTR\n");
                    printf("\tMOV SBUF,A\n");
                    printf("\tJNB TI, $\n");
                    printf("\tCLR TI\n");
                    printf("\tINC DPTR\n");
                    printf("\tDJNZ R1, LOOP\n\n");
                    printf("\tORG 100H\n");
                    printf("\tDB 'UNLOCK'\n");
                    printf("END\n\n");
                    break;

        default:    printf("Unknown command\n");
                    break;
    }
}
return 0;
}

```

build.sh

```

bison -y -d doorlock.y
flex doorlock.l
gcc -c y.tab.c lex.yy.c
gcc y.tab.o lex.yy.o doorlock.c -o doorlock_exec

```

clean.sh

```

rm doorlock_exec
rm lex.yy.o y.tab.o
rm lex.yy.c y.tab.c
rm y.tab.h

```

1) test results for few regular expressions

```

Terminal File Edit View Search Terminal Help
root@sohan-Inspiron-5547:/home/sohan/workspace/YaccandLex/src# ls -l
total 24
-rwxrwxr-x 1 sohan sohan 111 30 15:06 build.sh
-rwxrwxr-x 1 sohan sohan 70 30 15:06 clean.sh
-rw-rw-r-- 1 sohan sohan 1892 30 15:53 doorlock.c
-rw-rw-r-- 1 sohan sohan 768 29 12:36 doorlock.h
-rw-rw-r-- 1 sohan sohan 476 1 12:27 doorlock.l
-rw-rw-r-- 1 sohan sohan 2322 1 12:17 doorlock.y
root@sohan-Inspiron-5547:/home/sohan/workspace/YaccandLex/src# sh build.sh
root@sohan-Inspiron-5547:/home/sohan/workspace/YaccandLex/src# ls -l
total 196
-rwxrwxr-x 1 sohan sohan 111 30 15:06 build.sh
-rwxrwxr-x 1 sohan sohan 70 30 15:06 clean.sh
-rw-rw-r-- 1 sohan sohan 1892 30 15:53 doorlock.c
-rwxr-xr-x 1 root root 29112 1 13:35 doorlock_exec
-rw-rw-r-- 1 sohan sohan 768 29 12:36 doorlock.h
-rw-rw-r-- 1 sohan sohan 476 1 12:27 doorlock.l
-rw-rw-r-- 1 sohan sohan 2322 1 12:17 doorlock.y
-rw-r--r-- 1 root root 46085 1 13:35 lex.yy.c
-rw-r--r-- 1 root root 25360 1 13:35 lex.yy.o
-rw-r--r-- 1 root root 46113 1 13:35 y.tab.c
-rw-r--r-- 1 root root 2512 1 13:35 y.tab.h
-rw-r--r-- 1 root root 8728 1 13:35 y.tab.o
root@sohan-Inspiron-5547:/home/sohan/workspace/YaccandLex/src# ./doorlock_exec
door 2 unlock;

DATA P1 090H
DATA P2 0A0h
ORG 30H

        CLR P1.7
        MOV P2, #2
        MOV DPTR, #100h
        MOV R1, #11

LOOP:    CLR A
        MOVC A,@A+DPTR
        MOV SBUF,A
        JNB TI, $
        CLR TI
        INC DPTR
        DJNZ R1, LOOP

        ORG 100H
        DB '@01 RMOV -50'

END

door 3 lock;

DATA P1 090H
DATA P2 0A0h
ORG 30H

        CLR P1.7
        MOV P2, #3
        MOV DPTR, #100h
        MOV R1, #11

LOOP:    CLR A
        MOVC A,@A+DPTR
        MOV SBUF,A
        JNB TI, $
        CLR TI
        INC DPTR
        DJNZ R1, LOOP

        ORG 100H
        DB '@01 RMOV 50'

END

door 0 lock;

DATA P1 090H
DATA P2 0A0h
ORG 30H

        CLR P1.7
        MOV P2, #0
        MOV DPTR, #100h
        MOV R1, #11

LOOP:    CLR A
        MOVC A,@A+DPTR
        MOV SBUF,A
        JNB TI, $
        CLR TI
        INC DPTR
        DJNZ R1, LOOP

        ORG 100H
        DB '@01 RMOV 50'

END

```

```

check 0;

DATA P1 090H
ORG 30H

        SETB P1.7
        JB P1.0, LOCK

UNLOCK: MOV DPTR, #100H

        MOV R1, #6
        SJMP LOOP
LOCK:   MOV DPTR, #103H

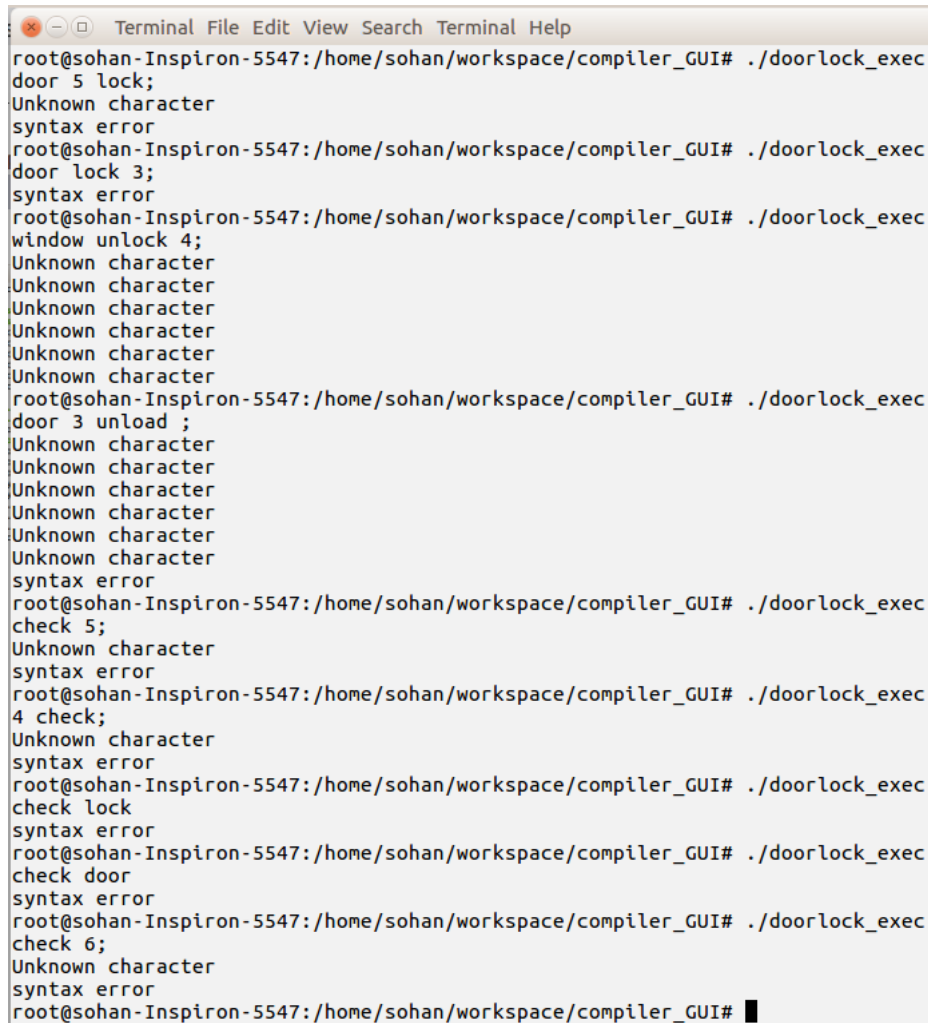
        MOV R1, #4
        SJMP LOOP
LOOP:   CLR A
        MOVC A, @A+DPTR
        MOV SBUF, A
        JNB TI, $
        CLR TI
        INC DPTR
        DJNZ R1, LOOP

        ORG 100H
        DB 'UNLOCK'

END

```

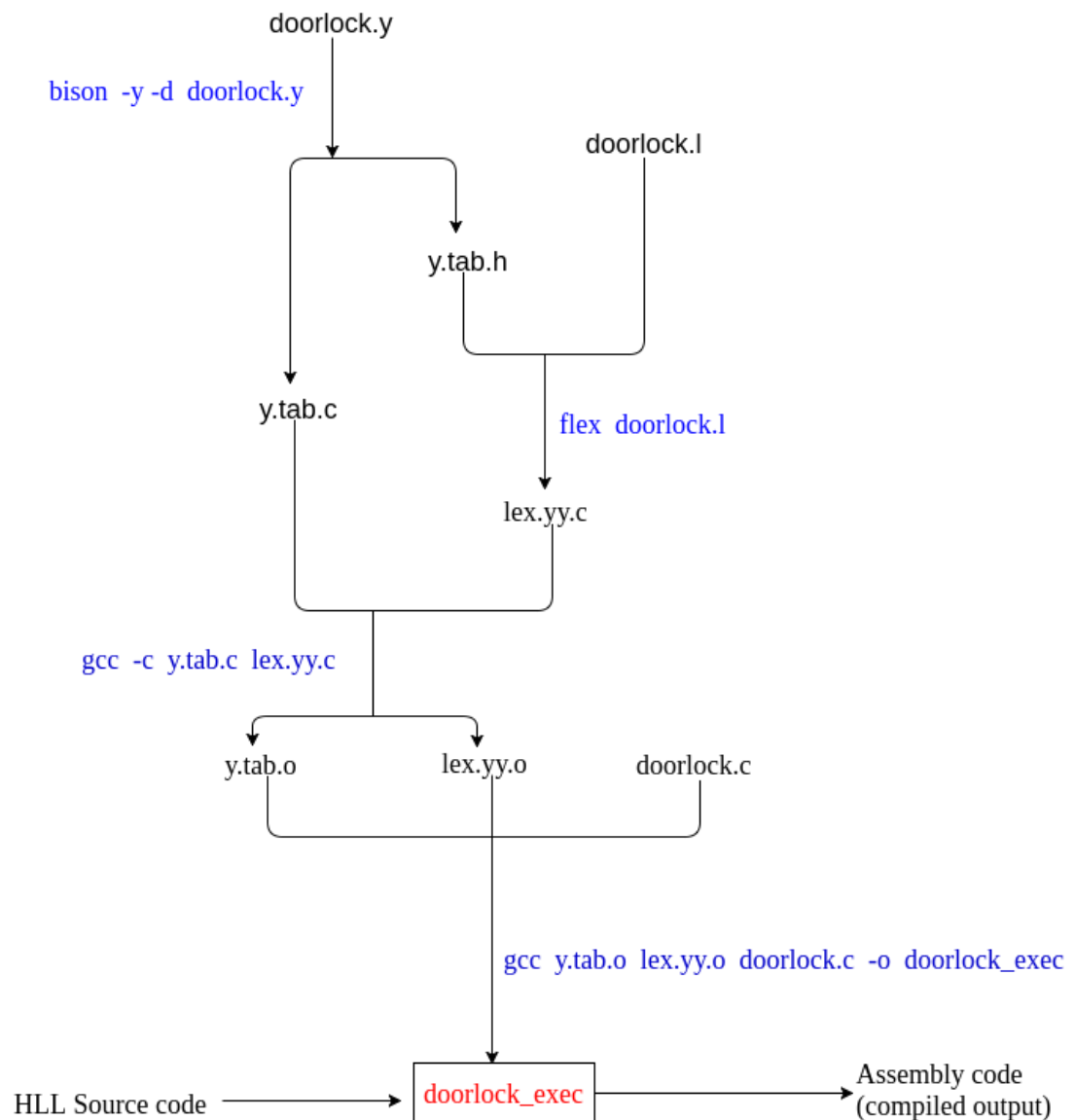
Error generation due to Incorrect syntax



```

Terminal File Edit View Search Terminal Help
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI# ./doorlock_exec
door 5 lock;
Unknown character
syntax error
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI# ./doorlock_exec
door lock 3;
syntax error
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI# ./doorlock_exec
window unlock 4;
Unknown character
Unknown character
Unknown character
Unknown character
Unknown character
Unknown character
Unknown character
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI# ./doorlock_exec
door 3 unload ;
Unknown character
Unknown character
Unknown character
Unknown character
Unknown character
Unknown character
syntax error
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI# ./doorlock_exec
check 5;
Unknown character
syntax error
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI# ./doorlock_exec
4 check;
Unknown character
syntax error
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI# ./doorlock_exec
check lock
syntax error
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI# ./doorlock_exec
check door
syntax error
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI# ./doorlock_exec
check 6;
Unknown character
syntax error
root@sohan-Inspiron-5547:/home/sohan/workspace/compiler_GUI#

```

Compilation is a process that translates a program in one language (the source language) into an equivalent program in another language (the object or target language).

The `doorlock.l` file reads the source program one character at a time and converts it into a stream of tokens. After tokenizing the input source code it generates the `lex.yy.c` file according to the program.

Then the `doorlock.y` takes the input from the `doorlock.l` file and generates a parse tree. In `doorlock.y` file defines the production rules and the actions for each tokens. When run the `doorlock.y` file it generates the `y.tab.c` and `y.tab.h` files. When compiling `lex.yy.c` and `y.tab.c` files using `gcc` compiler it generates the `lex.yy.o` and `y.tab.o` files. After that `lex.yy.o`, `y.tab.o` and `doorlock.c` compile using `gcc` and output the `doorlock_exec`. `doorlock_exec` is a executable file.

Interfacing GUI to the above executable (doorlock_exec) file.

I use java language to create the GUI

Java File

```
import java.awt.Color;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.ImageIcon;
import javax.swing.JButton;
import java.awt.event.ActionListener;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.awt.event.ActionEvent;
import javax.swing.JTabbedPane;
import javax.swing.JLabel;
import javax.swing.JEditorPane;

public class GUI extends JFrame {

    private static final long serialVersionUID = 1L;
    private JPanel contentPane;
    String door1_stat = "lock";
    String door2_stat = "lock";
    String door3_stat = "lock";
    String door4_stat = "lock";

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    GUI frame = new GUI();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }

    public GUI() {
        setResizable(false);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setBounds(100, 100, 692, 836);
        contentPane = new JPanel();
        contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
        setContentPane(contentPane);
        contentPane.setLayout(null);

        JTabbedPane tabbedPane = new JTabbedPane(JTabbedPane.TOP);
        tabbedPane.setBounds(23, 370, 637, 442);
        contentPane.add(tabbedPane);

        JEditorPane HLL_tab = new JEditorPane();
        tabbedPane.addTab("HLL", null, HLL_tab, null);

        JEditorPane Assembly_tab = new JEditorPane();
        tabbedPane.addTab("Assembly", null, Assembly_tab, null);
    }
}
```

```

        JButton btn_refresh_2 = new JButton("");
        btn_refresh_2
            .setIcon(new
ImageIcon(GUI.class.getResource("/com/sun/javafx/scene/web/skin/Redo_16x16_JFX.png"))
);
        btn_refresh_2.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                try {

                    BufferedWriter outfile = new BufferedWriter(new
FileWriter("input.txt"));
                    BufferedReader infile = new BufferedReader(new
FileReader("input.txt"));

                    outfile.write("check 1;");
                    outfile.close();

                    String line;
                    String mem = "";
                    while ((line = infile.readLine()) != null) {
                        mem = mem + line + "\n";
                    }
                    HLL_tab.setText(mem);
                    infile.close();

                    Process p = Runtime.getRuntime().exec("./doorlock_exec < input.txt
> output.txt");

                    p.waitFor();
                    p.destroy();

                    infile = new BufferedReader(new FileReader("output.txt"));
                    mem = "";
                    while ((line = infile.readLine()) != null) {
                        mem = mem + line + "\n";
                    }
                    Assembly_tab.setText(mem);
                    infile.close();

                } catch (Exception ex) {
                    System.err.println(ex);
                }

            }
        });

        btn_refresh_2.setBounds(596, 38, 51, 42);
        contentPane.add(btn_refresh_2);

        JButton btn_refresh_1 = new JButton("");
        btn_refresh_1
            .setIcon(new
ImageIcon(GUI.class.getResource("/com/sun/javafx/scene/web/skin/Redo_16x16_JFX.png"))
);
        btn_refresh_1.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {

                try {

                    BufferedWriter outfile = new BufferedWriter(new
FileWriter("input.txt"));
                    BufferedReader infile = new BufferedReader(new
FileReader("input.txt"));

                    outfile.write("check 0;");

```

```

        outfile.close();

        String line;
        String mem = "";
        while ((line = infile.readLine()) != null) {
            mem = mem + line + "\n";
        }
        HLL_tab.setText(mem);
        infile.close();

        Process p = Runtime.getRuntime().exec("./doorlock_exec < input.txt
> output.txt");

        p.waitFor();
        p.destroy();

        infile = new BufferedReader(new FileReader("output.txt"));
        mem = "";
        while ((line = infile.readLine()) != null) {
            mem = mem + line + "\n";
        }
        Assembly_tab.setText(mem);
        infile.close();

    } catch (Exception ex) {
        System.err.println(ex);
    }

    }
});
btn_refresh_1.setBounds(266, 38, 51, 42);
contentPane.add(btn_refresh_1);

JButton btn_refresh_3 = new JButton("");
btn_refresh_3
    .setIcon(new
ImageIcon(GUI.class.getResource("/com/sun/javafx/scene/web/skin/Redo_16x16_JFX.png"))
);
btn_refresh_3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        try {

            BufferedWriter outfile = new BufferedWriter(new
FileWriter("input.txt"));
            BufferedReader infile = new BufferedReader(new
FileReader("input.txt"));

            outfile.write("check 2;");
            outfile.close();

            String line;
            String mem = "";
            while ((line = infile.readLine()) != null) {
                mem = mem + line + "\n";
            }
            HLL_tab.setText(mem);
            infile.close();

            Process p = Runtime.getRuntime().exec("./doorlock_exec < input.txt
> output.txt");

            p.waitFor();
            p.destroy();

            infile = new BufferedReader(new FileReader("output.txt"));
            mem = "";
            while ((line = infile.readLine()) != null) {
                mem = mem + line + "\n";

```

```

        }
        Assembly_tab.setText(mem);
        infile.close();

    } catch (Exception ex) {
        System.err.println(ex);
    }

    });

    btn_refresh_3.setBounds(266, 186, 51, 42);
    contentPane.add(btn_refresh_3);

    JButton btn_refresh_4 = new JButton("");
    btn_refresh_4
        .setIcon(new
ImageIcon(GUI.class.getResource("/com/sun/javafx/scene/web/skin/Redo_16x16_JFX.png"))
);
    btn_refresh_4.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {

            try {

                BufferedWriter outfile = new BufferedWriter(new
FileWriter("input.txt"));
                BufferedReader infile = new BufferedReader(new
FileReader("input.txt"));

                outfile.write("check 3;");
                outfile.close();

                String line;
                String mem = "";
                while ((line = infile.readLine()) != null) {
                    mem = mem + line + "\n";
                }
                HLL_tab.setText(mem);
                infile.close();

                Process p = Runtime.getRuntime().exec("./doorlock_exec < input.txt
> output.txt");

                p.waitFor();
                p.destroy();

                infile = new BufferedReader(new FileReader("output.txt"));
                mem = "";
                while ((line = infile.readLine()) != null) {
                    mem = mem + line + "\n";
                }
                Assembly_tab.setText(mem);
                infile.close();

            } catch (Exception ex) {
                System.err.println(ex);
            }

        }

    });
    btn_refresh_4.setBounds(596, 186, 51, 42);
    contentPane.add(btn_refresh_4);

    JButton btn_door1 = new JButton();
    btn_door1.setText(door1_stat);
    btn_door1.setBackground(new Color(51, 204, 51));

```

```

btn_door1.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        if (door1_stat == "lock") {
            door1_stat = "unlock";
            btn_door1.setText(door1_stat);
            btn_door1.setBackground(new Color(255, 102, 102));
        } else {
            door1_stat = "lock";
            btn_door1.setText(door1_stat);
            btn_door1.setBackground(new Color(51, 204, 51));
        }

        try {

            BufferedWriter outfile = new BufferedWriter(new
FileWriter("input.txt"));
            BufferedReader infile = new BufferedReader(new
FileReader("input.txt"));

            outfile.write("door 0 " + door1_stat + ";");
            outfile.close();

            String line;
            String mem = "";
            while ((line = infile.readLine()) != null) {
                mem = mem + line + "\n";
            }
            HLL_tab.setText(mem);
            infile.close();

            Process p = Runtime.getRuntime().exec("./doorlock_exec < input.txt
> output.txt");

            p.waitFor();
            p.destroy();

            infile = new BufferedReader(new FileReader("output.txt"));
            mem = "";
            while ((line = infile.readLine()) != null) {
                mem = mem + line + "\n";
            }
            Assembly_tab.setText(mem);
            infile.close();

        } catch (Exception ex) {
            System.err.println(ex);
        }

    }
});
btn_door1.setBounds(36, 38, 216, 109);
contentPane.add(btn_door1);

JButton btn_door2 = new JButton();
btn_door2.setText(door2_stat);
btn_door2.setBackground(new Color(51, 204, 51));

btn_door2.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        if (door2_stat == "lock") {
            door2_stat = "unlock";
            btn_door2.setText(door1_stat);
            btn_door2.setBackground(new Color(255, 102, 102));
        } else {
            door2_stat = "lock";
            btn_door2.setText(door1_stat);

```

```

        btn_door2.setBackground(new Color(51, 204, 51));
    }

    try {
        BufferedWriter outfile = new BufferedWriter(new
FileWriter("input.txt"));
        BufferedReader infile = new BufferedReader(new
FileReader("input.txt"));

        outfile.write("door 1 " + door2_stat + ";");
        outfile.close();

        String line;
        String mem = "";
        while ((line = infile.readLine()) != null) {
            mem = mem + line + "\n";
        }
        HLL_tab.setText(mem);
        infile.close();

        Process p = Runtime.getRuntime().exec("./doorlock_exec < input.txt
> output.txt");

        p.waitFor();
        p.destroy();

        infile = new BufferedReader(new FileReader("output.txt"));
        mem = "";
        while ((line = infile.readLine()) != null) {
            mem = mem + line + "\n";
        }
        Assembly_tab.setText(mem);
        infile.close();

    } catch (Exception ex) {
        System.err.println(ex);
    }
}

});
btn_door2.setBounds(368, 38, 216, 109);
contentPane.add(btn_door2);

JButton btn_door3 = new JButton();
btn_door3.setText(door3_stat);
btn_door3.setBackground(new Color(51, 204, 51));
btn_door3.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        if (door3_stat == "lock") {
            door3_stat = "unlock";
            btn_door3.setText(door1_stat);
            btn_door3.setBackground(new Color(255, 102, 102));
        } else {
            door3_stat = "lock";
            btn_door3.setText(door1_stat);
            btn_door3.setBackground(new Color(51, 204, 51));
        }

        try {
            BufferedWriter outfile = new BufferedWriter(new
FileWriter("input.txt"));
            BufferedReader infile = new BufferedReader(new
FileReader("input.txt"));

            outfile.write("door 2 " + door3_stat + ";");
            outfile.close();

```

```

        String line;
        String mem = "";
        while ((line = infile.readLine()) != null) {
            mem = mem + line + "\n";
        }
        HLL_tab.setText(mem);
        infile.close();

        Process p = Runtime.getRuntime().exec("./doorlock_exec < input.txt
> output.txt");

        p.waitFor();
        p.destroy();

        infile = new BufferedReader(new FileReader("output.txt"));
        mem = "";

        while ((line = infile.readLine()) != null) {
            mem = mem + line + "\n";
        }
        Assembly_tab.setText(mem);
        infile.close();

    } catch (Exception ex) {
        System.err.println(ex);
    }

    }
});
btn_door3.setBounds(36, 186, 216, 109);
contentPane.add(btn_door3);

JButton btn_door4 = new JButton();
btn_door4.setText(door4_stat);
btn_door4.setBackground(new Color(51, 204, 51));
btn_door4.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {

        if (door4_stat == "lock") {
            door4_stat = "unlock";
            btn_door4.setText(door4_stat);
            btn_door4.setBackground(new Color(255, 102, 102));
        } else {
            door4_stat = "lock";
            btn_door4.setText(door4_stat);
            btn_door4.setBackground(new Color(51, 204, 51));
        }

        try {

            BufferedWriter outfile = new BufferedWriter(new
FileWriter("input.txt"));
            BufferedReader infile = new BufferedReader(new
FileReader("input.txt"));

            outfile.write("door 3 " + door4_stat + ";");
            outfile.close();

            String line;
            String mem = "";
            while ((line = infile.readLine()) != null) {
                mem = mem + line + "\n";
            }
            HLL_tab.setText(mem);
            infile.close();

```



```

        Process p = Runtime.getRuntime().exec("./doorlock_exec < input.txt
> output.txt");

        p.waitFor();
        p.destroy();

        infile = new BufferedReader(new FileReader("output.txt"));
        mem = "";
        while ((line = infile.readLine()) != null) {
            mem = mem + line + "\n";
        }
        Assembly_tab.setText(mem);
        infile.close();

    } catch (Exception ex) {
        System.err.println(ex);
    }

    }
});
btn_door4.setBounds(368, 186, 216, 109);
contentPane.add(btn_door4);

JLabel doorlbl_1 = new JLabel("Door 1");
doorlbl_1.setBounds(109, 159, 70, 15);
contentPane.add(doorlbl_1);

JLabel doorlbl_2 = new JLabel("Door 2");
doorlbl_2.setBounds(448, 159, 70, 15);
contentPane.add(doorlbl_2);

JLabel doorlbl_3 = new JLabel("Door 3");
doorlbl_3.setBounds(109, 307, 70, 15);
contentPane.add(doorlbl_3);

JLabel doorlbl_4 = new JLabel("Door 4");
doorlbl_4.setBounds(448, 307, 70, 15);
contentPane.add(doorlbl_4);

}
}

```

Java Swing GUI Integration with the compiler

