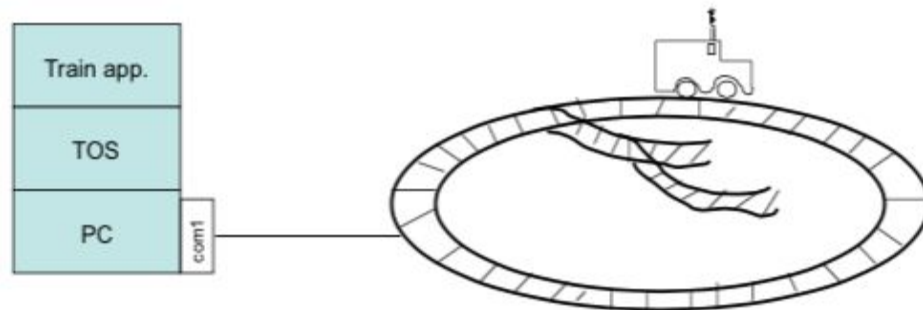


TOS Development

Train Setup



Goal:

- To run the train application through the shell.
- The abandoned wagon must be collected by the train and return to home base without colliding with the Zamboni if it exists.
- Shell must perform basic functions like clear window, print current processes, etc.

Steps:

- Implement the assignments given to set the base functions required for shell and train application.
- Run the test cases in ttc.jar to see if test is successful
- The order of implementation was:
 - String length, copy and compare functions in stdlib.c
 - Peek and Poke functions in mem.c
 - Tos windows functions in window.c
 - Create process and print process functions in process.c
 - Scheduling functions in dispatch.c
 - resign() for context switching in dispatch.c
 - Functions for interrupts in intr.c
 - isr_timer() for pre-emptive scheduling in intr.c
 - wait_for_interrupt() for synchronizing interrupts in intr.c
 - Data communication functions in com.c
 - TOS shell functions in shell.c
 - Train application functions in train.c
 - Functions for pacman in pacman.c

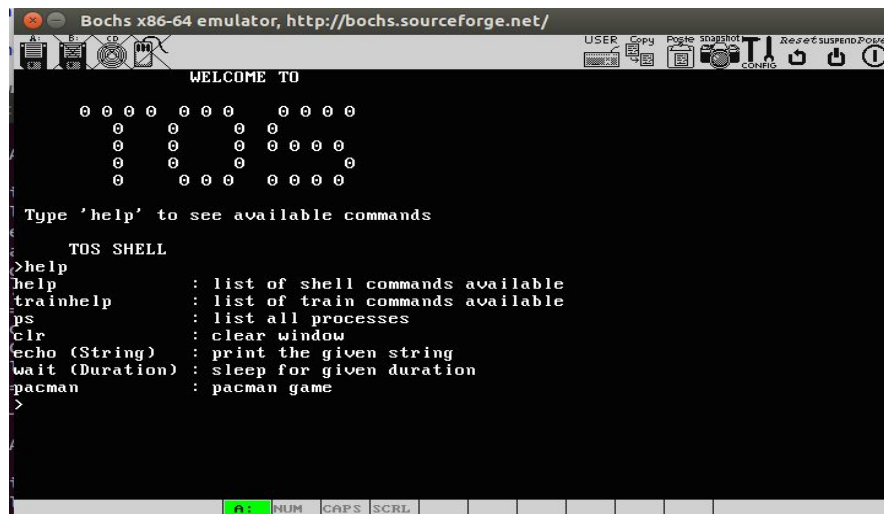
Challenges:

- Working on Virtual Machine.
- Lack of expertise in C programming.
- Difficult to debug.
- Manually getting the abandoned wagon in the simulator was easy, but making it work automatically from the shell required a lot of testing.

Result:

Functions implemented in TOS and their corresponding output are shown below:

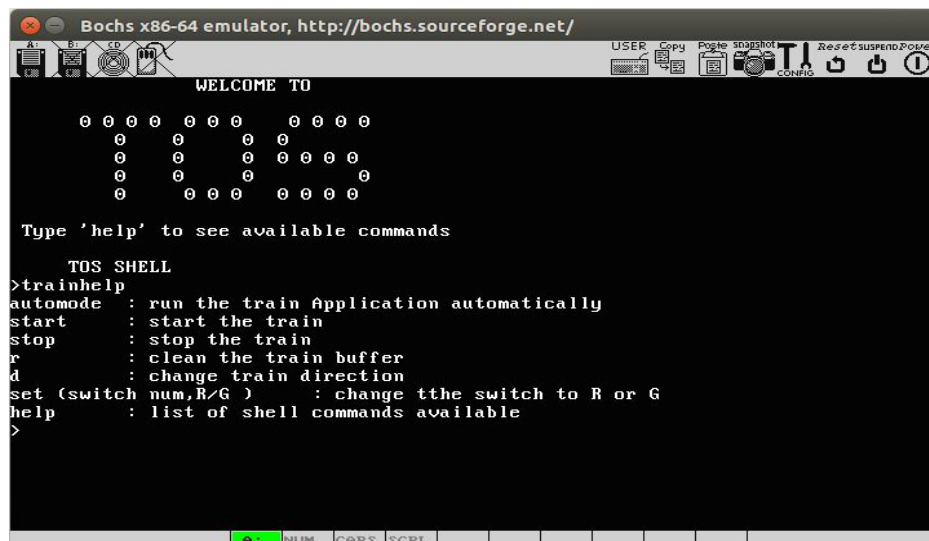
- Divided the TOS shell window into three sections, for Train, Shell and Pacman.
- **printHelp()** prints the Shell commands that are available.



```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
WELCOME TO
  0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0
          0 0 0 0 0 0
Type 'help' to see available commands
TOS SHELL
>help
help          : list of shell commands available
trainhelp     : list of train commands available
ps            : list all processes
clr           : clear window
echo (String) : print the given string
wait (Duration) : sleep for given duration
pacman        : pacman game
>
```

Figure 1: TOS Shell help commands

- **printTrainHelp()** prints the available train commands.



```
Bochs x86-64 emulator, http://bochs.sourceforge.net/
WELCOME TO
  0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0
      0 0 0 0 0 0 0 0
        0 0 0 0 0 0 0
          0 0 0 0 0 0
Type 'help' to see available commands
TOS SHELL
>trainhelp
automode      : run the train Application automatically
start         : start the train
stop          : stop the train
r             : clean the train buffer
d             : change train direction
set (switch num,R/G ) : change tthe switch to R or G
help         : list of shell commands available
>
```

Figure 2: Train help commands

- `printAllProcs()` prints all process

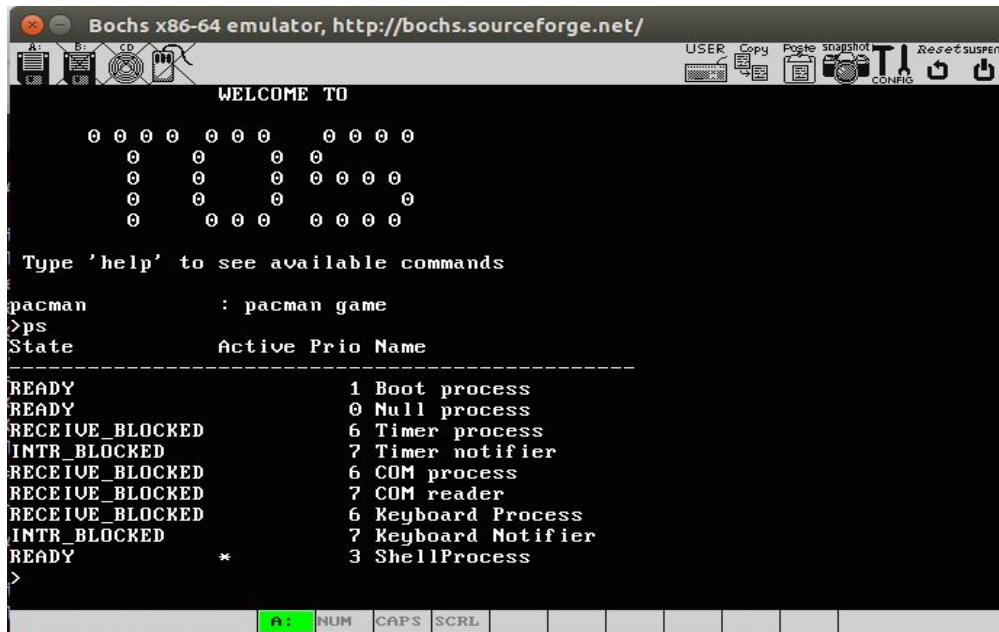


Figure 3: List of processes

- `echoStr()` echo the string entered in the shell

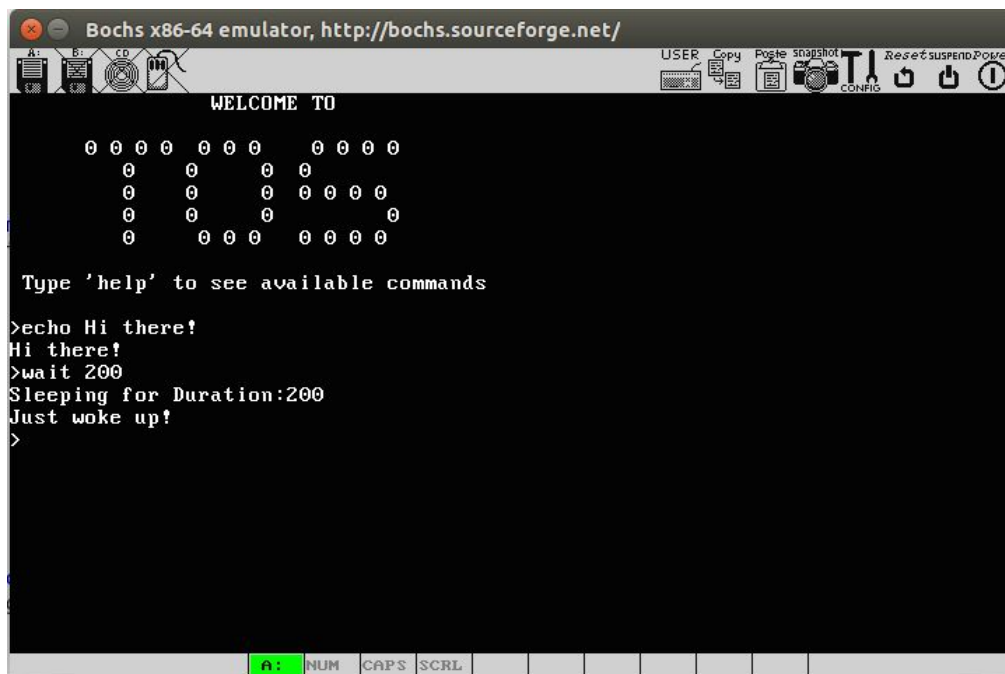


Figure 4: output of echo and wait functions

- `waitfor()` sleeps the shell for the given time
- `clearShellWin()` clears the shell window

- `runPacman()` initializes pacman process



Figure 5: Pacman game

Train functions

- `automode()` runs the train automatically with selected configuration

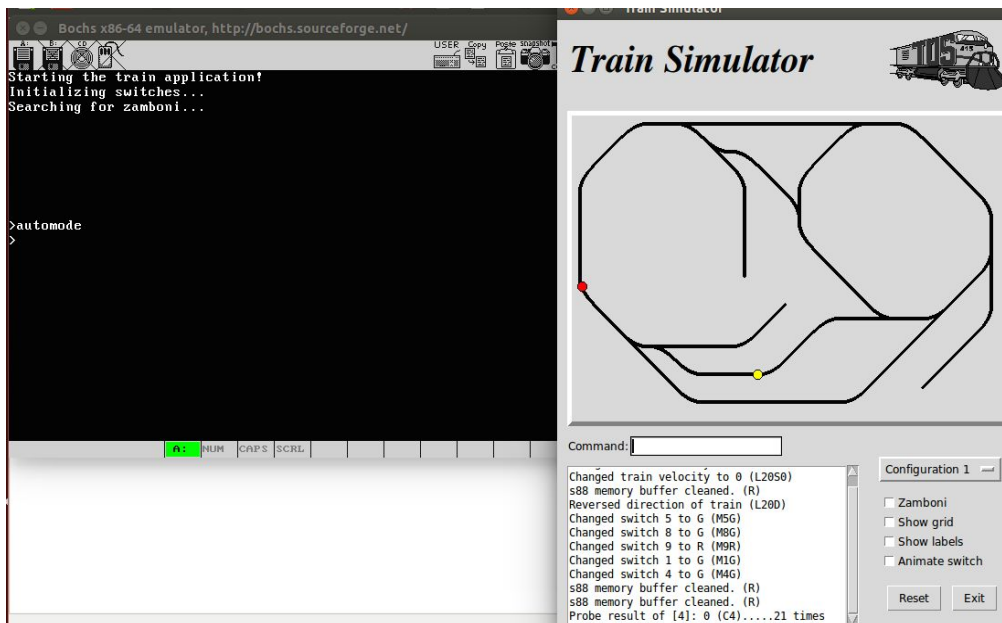


Figure 6: shows train running in configuration 1 automatically

- **trainStart()** starts the train on the simulator by setting train speed = 4

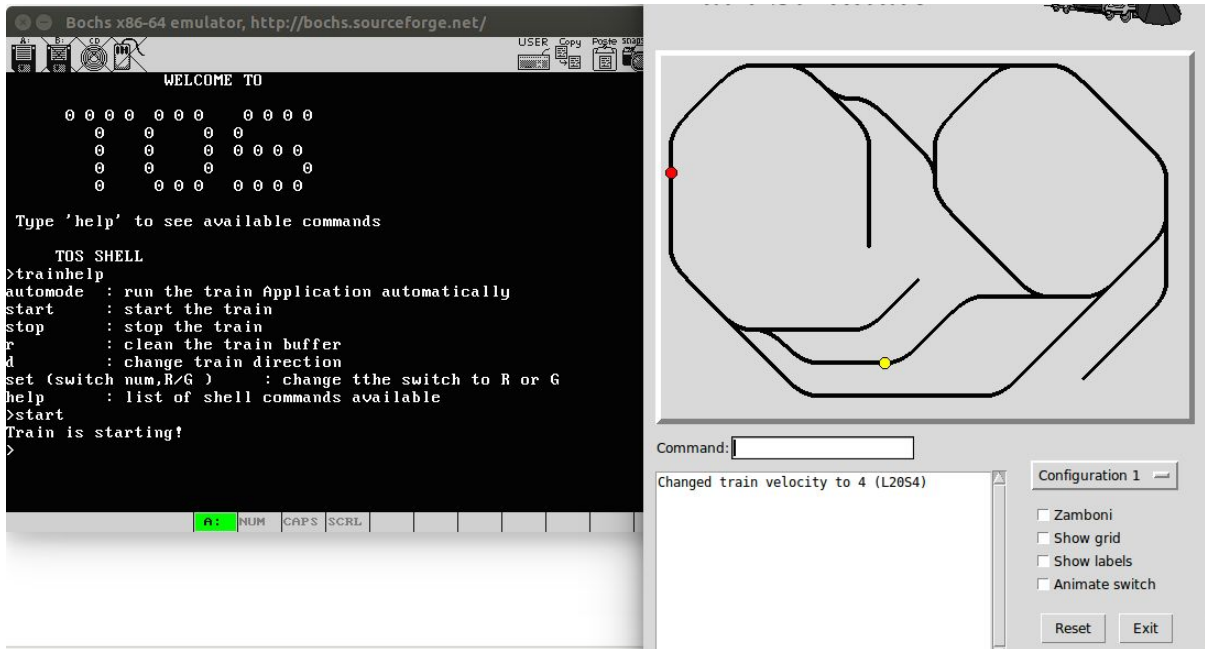


Figure 7: start the train

- **trainStop()** stops the train on the simulator by setting train speed = 0

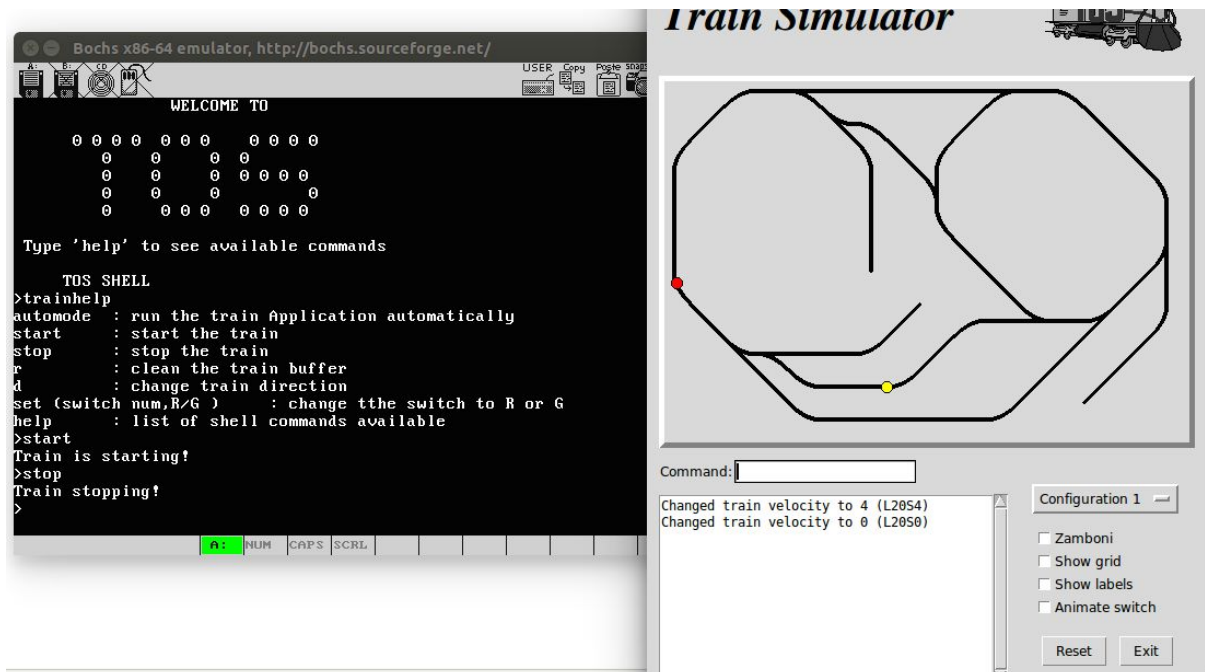


Figure 8: stop the train

- `cleanBuffer()` clears S88 buffer for train

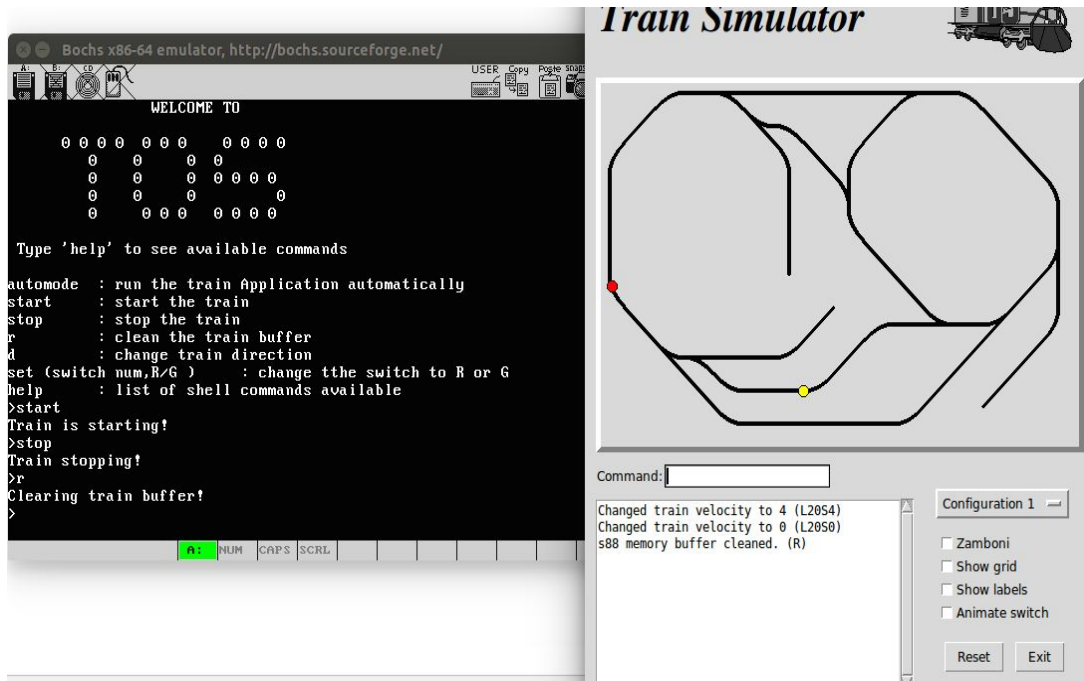


Figure 9: clears the train buffer

- `changeDirection()` changes train direction

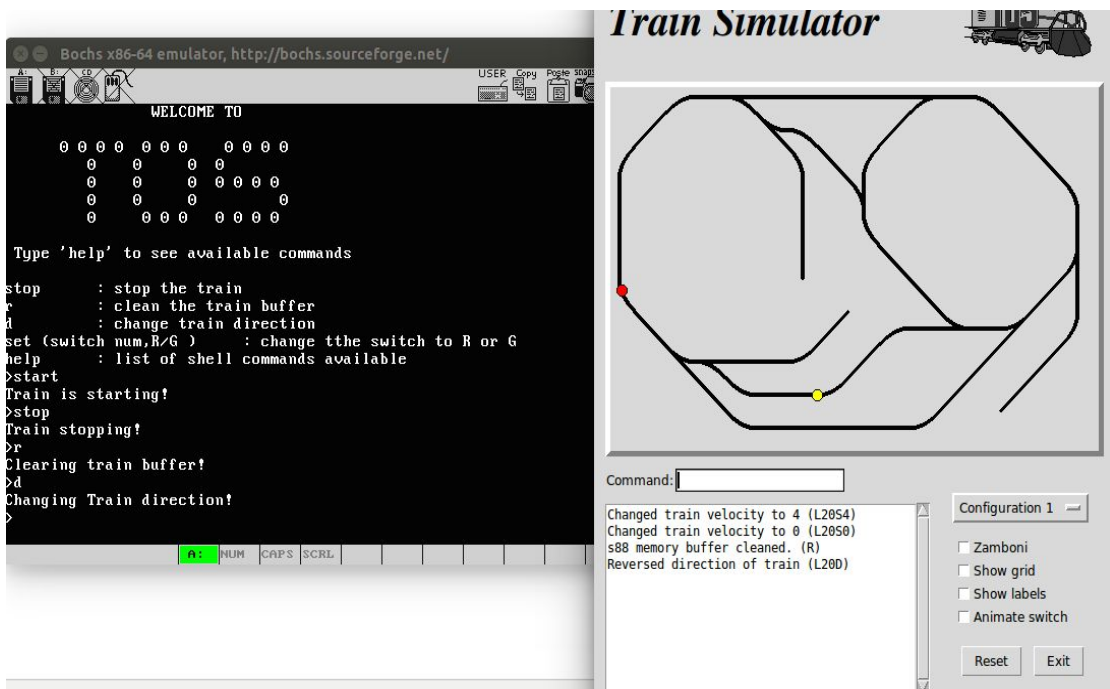


Figure 10: change the direction of the train

- Train Simulator

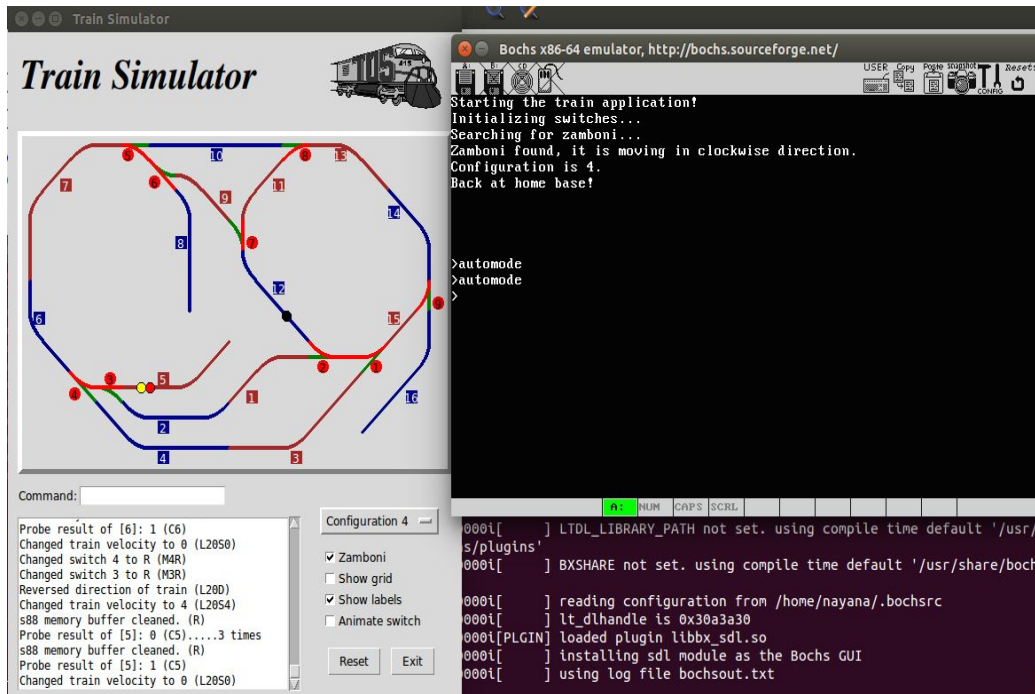


Figure 11: train running on configuration 4 with Zamboni