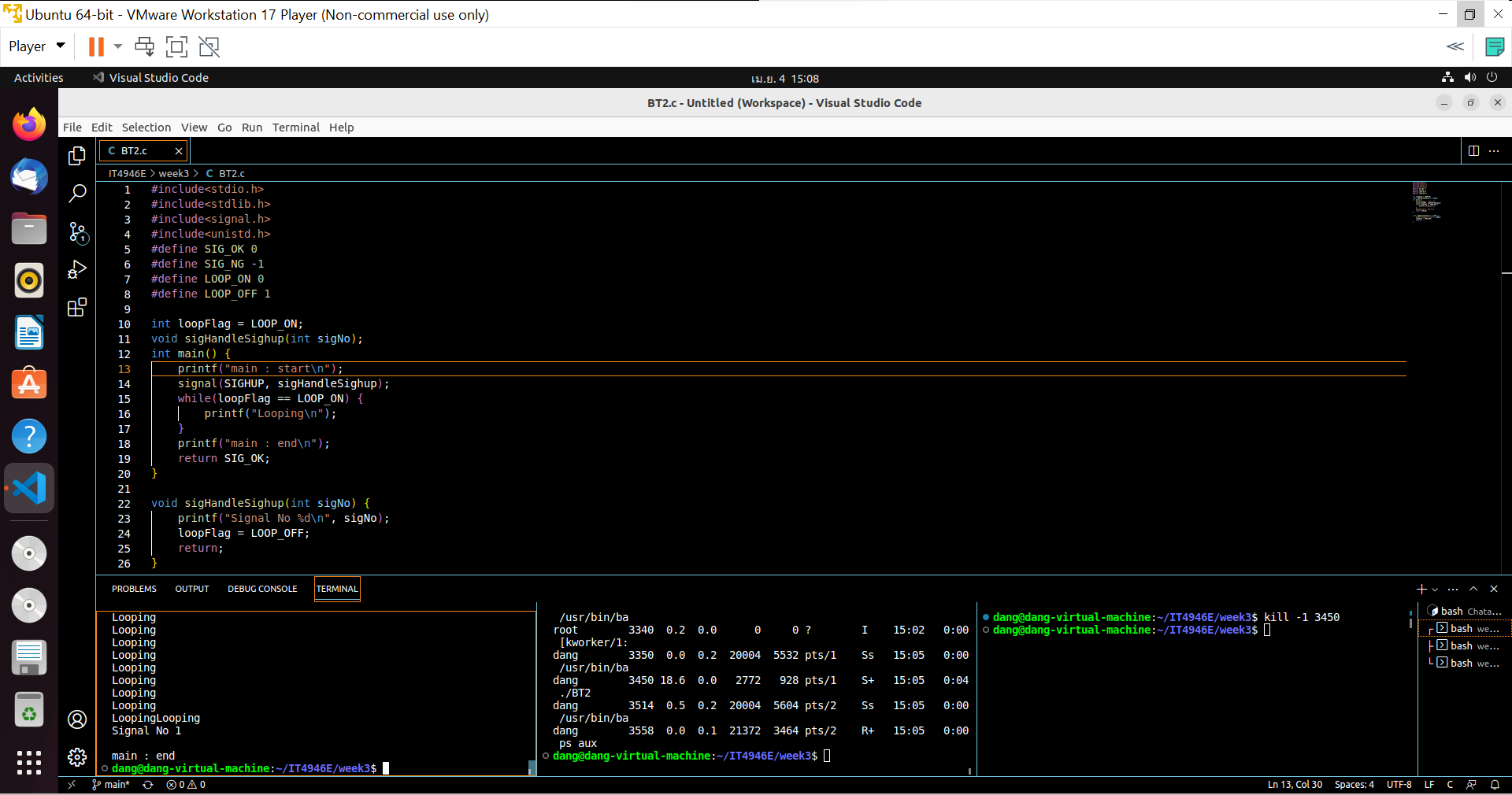
**Lab 3 Report**

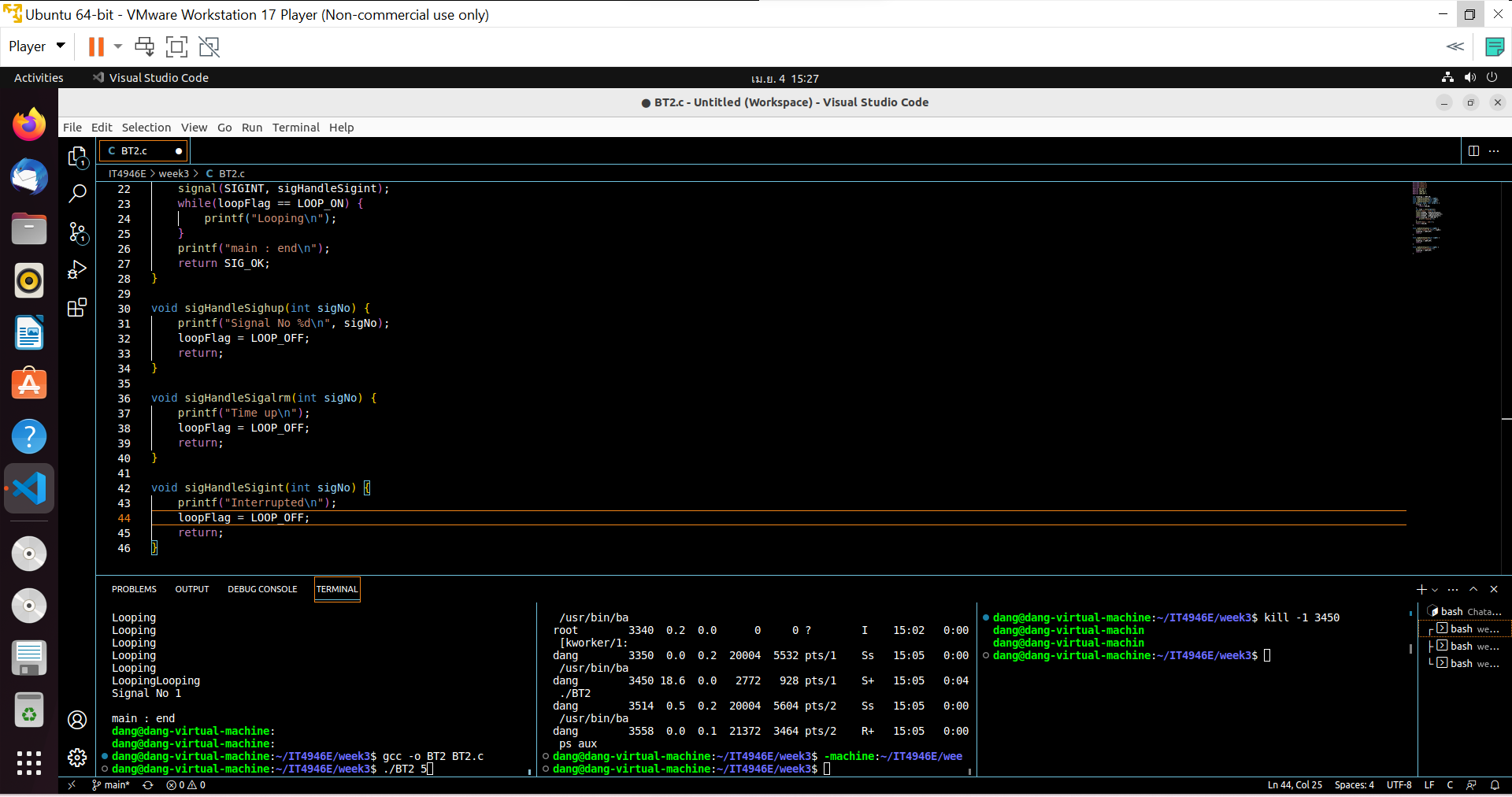
**Pham Hai Dang – 20194736**

1)

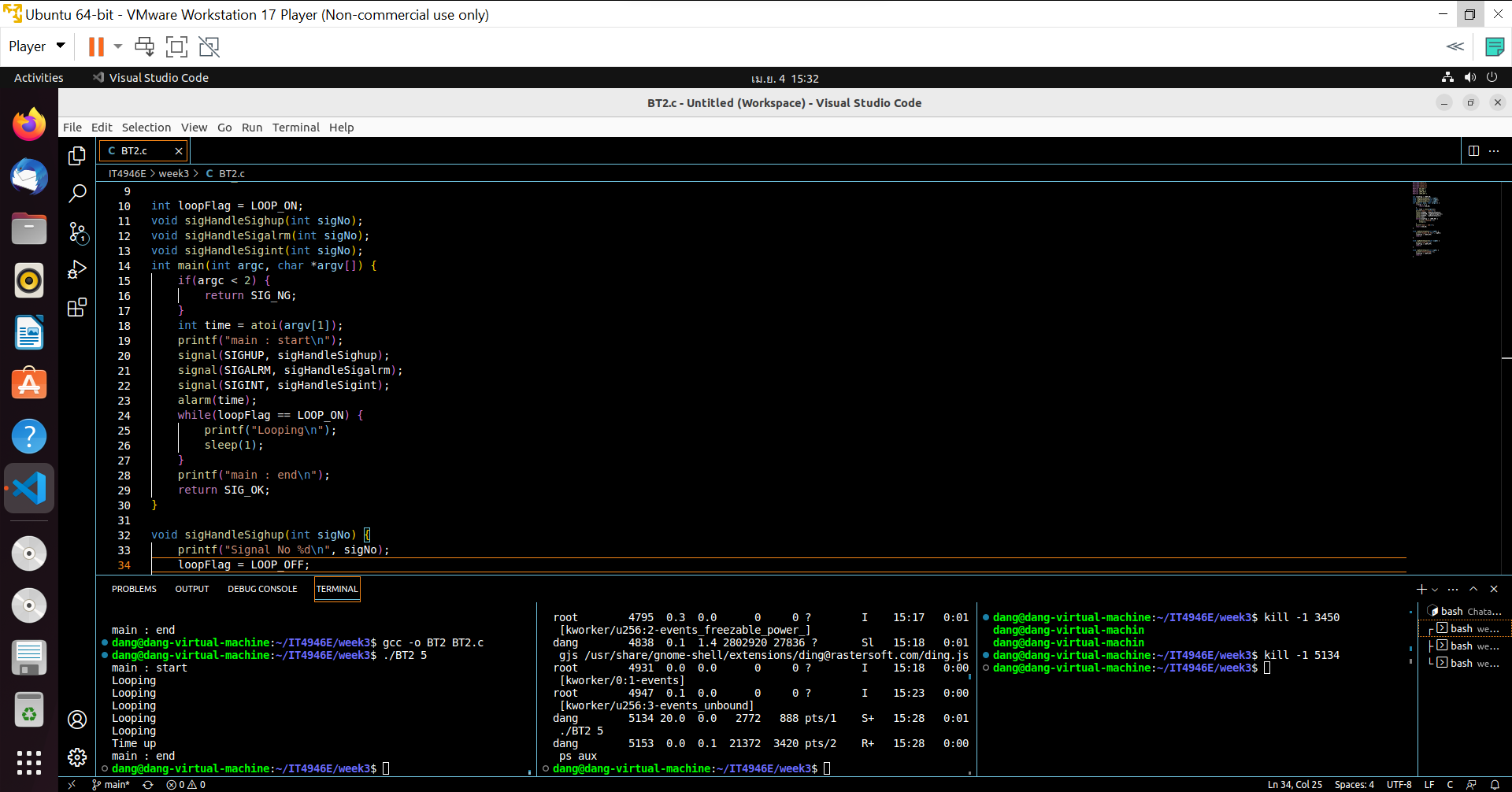


After running, I print the processID as you see here is 3450. After that, I kill that process and the main program stop and print ‘main : end’

2)



I add more two function to control timeup and interrupted as in image



The time here is 5 second and after 5 second, the program stop

A screenshot of a computer

Description automatically generated with medium confidence

When program is running, I press Ctrl + C and the program is interrupted

Explain: The program takes an argument **time** from the command line, which is used to set an alarm signal using the **alarm()** function. The **while** loop keeps running until a signal is received that sets the **loopFlag** to **LOOP\_OFF**, at which point the loop exits and the program terminates.

The program sets up signal handlers for three signals:

* **SIGHUP**: This is sent to the process when the controlling terminal is closed or the process is detached from it. The signal handler **sigHandleSighup()** sets the **loopFlag** to **LOOP\_OFF** to stop the program.
* **SIGALRM**: This is sent to the process when the **alarm()** timer expires. The signal handler **sigHandleSigalrm()** sets the **loopFlag** to **LOOP\_OFF** to stop the program.
* **SIGINT**: This is sent to the process when the user presses the **Ctrl+C** key combination. The signal handler **sigHandleSigint()** sets the **loopFlag** to **LOOP\_OFF** to stop the program.

When a signal is received, the corresponding signal handler is executed, which sets the **loopFlag** to **LOOP\_OFF** to stop the program. The **while** loop checks the value of **loopFlag** in each iteration, and when it is set to **LOOP\_OFF**, the loop exits and the program terminates.