CS3340 – Computer Architecture Final Project

Team:

Rahat Ahmed

* MIDI file import and parsing

Jonathan Darling

* Song playback

Kapoor Sookoo

* MIDI Piano (extra credit)

Issues:

* Analysis of test files to understand the MIDI file format
  + Solved by finding old command line MIDI tools, running them under a DOS emulator, and observing the results.
* Storing of MIDI events in a meaningful way.
  + Solved by implementation of a linked list to store the events for efficient playback.
* Conversion of delta ticks to milliseconds.
  + Solved by approximating an appropriate scale to convert between the two, and using Omar Darwish’s Mars4MIDI modification of the MARS IDE in order to have note durations play correctly.
* Lack of sufficient system calls in existing Mars simulator.
  + Solved by using the modified Mars simulator, Mars++.
* Distinguishing between the various MIDI commands.
  + Solved by implementing a large “switch” statement” to distinguish between the commands.
* Midi events with running status
  + Solved by rearranging file parsing logic to account for events that do not explicitly give a command byte.
* I/O Polling
  + Solved by using the Keyboard and Display simulator
* Converting type 1 midi songs to type 0
  + Using a midi command line tool in a DOS emulator

What each team member has learned:

Jonathan Darling:

This project really opened my eyes to how everything can be represented as data. I have always had a hard time grasping how something such as audio or video can be stored as 1s and 0s. Working with the MIDI format has given me a deep insight into how this process works. Through my implementation of a complex switch statement, I have gained an understanding of how logical operations can be designed in a low level language such as MIPS assembly.

Rahat Ahmed:

I have gained a deep understanding of the MIDI file format, including it’s events and quirks like running status. Tricks like running status were tricky to implement but necessary for shrinking files back when storage space was precious. Furthermore, I learned how to implement a linked list in assembly, which was surprisingly easier than it is in higher level languages. I learned that commenting is very important for writing MIPS code, because trying to understand code you wrote a long time ago is difficult without an explanation. Piazza was a great place to have questions answered, and I feel I did my part in answering questions myself. The programs uploaded there were definitely helpful for everybody. And most importantly of all, I now appreciate how easy it is to manage data and program logic in higher-level languages.

Kapoor Sookoo: