

I1 実験レポート

03190449 堀紡希

6月24日

1 I1 実験で学んだこと

I1 実験においてまず学んだことはコマンドラインの扱いと Emacs である。コマンドは既知のものが多かったが”ls”が”ls <カレントディレクトリ>”の略であるということは初めて知った。

Emacs は普段 VisualStudioCode を使っているのでなかなか慣れることができなかったが gud-gdb によるデバッグは非常に効果的で、バグを取り除く課題も gud-gdb を使うことで容易にクリアすることができた。

次に学んだのはバイナリファイルの読み書きである。

これまで標準入力、あるいはファイルに対して数字や文字列を人間が読めるような形で扱ってきたのでコンピュータにとって扱いやすい形であるバイナリ形式のデータを扱ったのは初めてのことであった

これは read, write 関数を用いると扱うことができる。どのデータを入力して、何が返ってくるのか、なれるまでは少し混乱した

デバッグの使い方も勉強になった。これまで printf でどこがおかしいのか探していたのでデータがどのように流れているのかわかりやすいデバッグはとても役に立った。

これに関連してエラー処理も学んだ。プログラミング基礎演習で少し Python のエラー処理は学んだが、ここでそのエラーが起こった時点でプログラムを停止させるという重要性を改めて学んだ。

次に本課題のメイン部分である音の処理のコマンドを学んだ。まずそれぞれのデータの形式、wav, raw, dat, ogg がどのように規定されているのかを学び、そしてそれらを rec, play を用いて録音、再生した。しかしその後 sox コマンドを学ぶと rec も play も sox の特殊例であることがわかり、音を様々に処理し、扱うことができるようになった。また sox コマンドを用いて wav から raw、dat から wav というようにデータの形式もできるようになった。

パイプライン、リダイレクトもとても重要であった。パイプラインを用いると直接プログラム同士を結びつけることができ、リダイレクトは標準入出力先をファイルにすることができる。また、ターミナルで長いコマンドを実行するためにシェルスクリプトの使い方を学習した。これはコマンドが長くなりがちな sox コマンドを扱うとき、特に細かく修正したりするときに重宝した。

データを可視化してくれる gnuplot, audacity の扱い方も学んだ。

最後に信号処理工学で扱った高速フーリエ変換を実装し、思うような周波数帯の信号を取り出し、切り取ることができるようになった。例えば dq1.wav (ドラゴンクエストの音声) を 1000Hz でローパスすると特徴的な音が聞こえなくなり、わかりやすくローパスフィルタを実現することができた。