

## I2 情報第2部(考察レポート)

工学部電気電子工学科  
学籍番号 03-190503 西山 晃人

### 1 考察

#### 1.1 接続してきたクライアントには、接続してきた時点以降サーバ付近でなっていた音のデータが送られるようにする

サーバ側で以下のようなコマンドでサーバを立ち上げ、rec で録音したデータをサーバで読み込むようにすると

```
$ rec -t raw -b 16 -c 1 -e s -r 44100 - | ./serv_send ポート番号
```

クライアントで接続してそこで鳴る音はサーバをたてた時点からの音になってしまう。それは、上記のコマンドを打つとサーバが accept で待っていてかつ rec 自身は録音を開始している状態となるからである。そこで popen というライブラリ関数(ソースコードによってターミナルでのコマンドを実行することができる)を使ってその問題の解決を図った。C 言語のコード内で以下のように記述するとソースコードが上から実行され、これが記述された行に到達するとターミナル内で rec が実行される。(die は自作の関数で popen が適切になされないと error を検出してプログラムを終了することができる。)

```
#include <stdio.h>

FILE *p;
if((p=popen("rec -t raw -b 16 -c 1 -e s -r 44100 -", "r"))==NULL) die("popen");
```

また rec で録音されたデータはファイルポインタ p でさされた場所に格納されることに注意する必要がある。これにより、ソースコードの任意の場所で rec を実行することで、任意のタイミングで録音することができる。

#### 1.2 双方向通信の実現

次はサーバとクライアントで双方向通信ができるようにプログラムを作成する。

サーバ

```
$ rec -t raw -b 16 -c 1 -e s -r 44100 - | ./serv_send ポート番号 | play -t raw
-b 16 -c 1 -e s -r 44100 -
```

クライアント

```
$ rec -t raw -b 16 -c 1 -e s -r 44100 - | ./serv_send サーバアドレス ポート番号
| play -t raw -b 16 -c 1 -e s -r 44100 -
```

上記のコマンドは実験教科書に記述されていたものでこれをサーバ、クライアント側で実行すると 1 秒経たない間に固まってしまう、プログラムが正常に動作しない。従来のソースコードでは while 文の中に recv と send を記述していた。この方法の場合、send と recv が交互に呼ばれ順番に実行されているため rec で録音される速度に対して while 文内での処理速度が十分でない、recv と send の処理でのメモリ領域でデータハザードを起こしているなどの問題点が考えられる。そこで pthread というライブラリを用いて並列処理をすることにより、応答速度を改善し、メモリ領域での競合をなくすことを考え、以下のように記述した。ここで exe\_recv、exe\_send は自作の関数であり、内部で while 文を回すことにより recv と send をプログラム終了時まで行なっている。

```
#include <pthread.h>

pthread_t pthread;
pthread_create(&pthread, NULL, &exe_recv, &s);
exe_send(s);
pthread_join(pthread, NULL);
```

このように記述し並列処理を行うことにより約 5 分ほどの通話が可能であった。さらに長時間の通話は試していないが十分長い時間の通話が可能であると考えられる。

## 2 参考文献

- [1] 電子情報工学科・電気電子工学科、「電気電子情報第一 (前期) 実験」、東京大学工学部 (2019)