

仕様書及び説明書

法政大学大学院 理工学研究科電気電子工学専攻 修士 2 年

澤田亮太

作成したプログラム (pro.c) の仕様について、これは入力ファイルを“log_ex1.txt”とし、監視ファイルより得られたデータを基に、計算結果をターミナルに出力するものとなっている。なお開発環境は VSCode、コンパイルコマンドは gcc である。実行結果のターミナルの出力のスクリーンを切り取ったものを図 1 に示す。

```
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.

C:\Users\chala\Documents\vscode\pro_fixpoint>cd "c:\Users\chala\Documents\vscode\pro_fixpoint\" && gcc -fexec-charset=CP932 pro.c -o pro && "c:\Users\chala\Documents\vscode\pro_fixpoint\"pro

---error detected!---
server:10.20.30.1/16
TBF(y-m-d)=(0000/00/00)
(h:m:s)=(09:59:00)

---error detected!---
server:192.168.1.2/24
TBF(y-m-d)=(0000/00/00)
(h:m:s)=(00:10:00)

---server overload detected---
server :10.20.30.1/16
before      till
(y-m-d)=(2020/10/19)~(2020/10/19)
(h:m:s)=(13:33:24)~(23:42:26)

---server overload detected---
server :192.168.1.2/24
before      till
(y-m-d)=(2020/10/19)~(2020/10/19)
(h:m:s)=(13:32:34)~(23:31:34)

---server overload detected---
server :192.168.1.2/24
before      till
(y-m-d)=(2020/10/19)~(2020/10/19)
(h:m:s)=(23:32:34)~(23:42:36)

C:\Users\chala\Documents\vscode\pro_fixpoint>
```

図 1 解析結果例

以下に、プログラムの行数及び補足事項を示す。

- ✓ 行 1~15 c 言語の動作に必要とする標準関数及びパラメータ。txt_name は、文字列 char 型のサイズの上限值であり、主に読み取る監視ファイルの文字数に影響する。また本プログラムではアドレスを文字列として格納しているため、この文字上限

にも影響する。max_server の値を調整することで、同時に監視できるサーバーアドレスの数の増減が可能である。なお、この値を超過するサーバー数が検知された時の動作は未設定。

- ✓ 行 16～67 本プログラムの動作に用いる変数の宣言及び初期化。文字列を格納する箇所については意識的に同定確保するようにしているが、その限りではない箇所も存在する。監視ログより読み取った日時をそれぞれ分けて格納している。末尾の err は、故障時 error にて用いることを意味している。
- ✓ 行 69 監視ログファイルの読み込み及び、読み込み失敗時の動作を記している。
- ✓ 行 75～107 読み込んだ情報を保存している。タイムアウトを示す'-'は一度文字 char 型として保存した後、例外的に ping を 10^6 として保存している。また保存されていないサーバーアドレスの格納先は"Not assigned"で初期化している。
- ✓ 行 109～129 タイムアウトを検知した時の動作である。初めてタイムアウトを検知したときのみその日時を保存している。
- ✓ 行 130～183 タイムアウトを検知しなかったときの動作である。パラメータとして設定した回数、タイムアウトが連続したかの判定を行った後、その有無に関わらず buffer_times の初期化を行っている。先ほど保存した、タイムアウト発生時の日時との差を計算しているが、この計算方法では、1 か月以上の計算結果が誤る k とが予想される。また閏年の考慮はされていないので、今後の更新課題である。
- ✓ 行 184 過負荷検知に用いる変数の宣言。
- ✓ 行 187～197 変数 overload 内にて、使用するデータの選択を行う。不要な部分は-1を設定し、のちの計算で除外する。また保存した日時のデータのフヨウな部分は全て(0000/00/00 00:00:00)と設定した。
- ✓ 行 199～220 平均応答時間を算出し、過負荷状態か変数 overload に情報を送る。なお平均応答時間の処理について、処理の誤った部分が存在し、1/m 倍しただけの計算となっている（例えば1つ目のデータのみを参照した場合、本ケースでは平均応答時間は2ms となるのが正しいが、2/3ms が計算結果として出力される）。
- ✓ 行 222～287 故障検知プログラムと同手法で日時の計算を行っている。なお末尾の be は、過負荷検知開始時 before を表している。変数 overload 内の判断結果を満たし、過負荷状態を検知したときのみ、保存していたデータを日時 before に代入しており、それ以外では初期化しているため、この初期化の有無によって、過負荷状態が終了したかどうか判別している。