

J3 Network Exercise

#01 04/07 (Thu) 10:35-12:05

Yasuyuki SAITO

National Institute of Technology, Kisarazu College



■ この授業で扱う主な内容

- C言語によるソケット通信のプログラミング。
- 画面上の文字列の操作。
- C言語の基本。
 - ポインタは理解している・・・かな?
 - Makefile の使い方は?
 - かつて、2年生のプログラミング授業を担当したときに作成した資料を置いておくので、復習・理解しておいて下さい。
 - ※C言語プログラミングの資料の「レポート課題」は、やらなくて結構。

今回のミッション

2

■ 2年生向けのC言語プログラミングの資料を熟読

- Teams 上にアップしました。0434_ネットワーク演習のチャンネルの「ファイル」からたどってください。

■ 教科書 第1章を熟読

- 当座、現時点では20%くらい理解できればOK。

■ キーワード

- パケット
- プロトコル
- ポート番号
- クライアント
- サーバ
- ソケット

今回のミッション

3

- 教科書 第1章を熟読(続き; 読み進めていく上での必要事項)
 - ネットワークは、プロトコル(通信規約)によって定義されている。
 - ネットワークプロトコルは、いくつかの階層に分かれている。
 - 基本的なプロトコルの「OSI参照モデル」
 - 第7層 アプリケーション層
 - 第6層 プレゼンテーション層
 - 第5層 セッション層
 - 第4層 トランスポート層
 - 第3層 ネットワーク層
 - 第2層 データリンク層
 - 第1層 物理層
 - TCP/IP とは、トランスポート層の「TCP」と、ネットワーク層の「IP」を使うということを意味する。
 - Transmission Control Protocol
 - Internet Protocol

今回のミッション

4

- 教科書 第1章を熟読(続き; 読み進めていく上での必要事項)
- プロトコルが階層化されているのは、ある定義の修正や追加の影響を、その階層の範囲(とその上下の層にも少し関連)にだけに限定できる点が優れている。
- もしも、少しの修正によってプロトコル全体をすべて見直す必要があるという
と、影響が大きすぎる。
- 新しいプロトコルを開発したとき、既存の規格と置き換えやすい。

今回のミッション: 演習の準備

5

■ 学外接続サーバ cs.jga.kisarazu.ac.jp に接続

■ SSH2 クライアントの例 (あくまでも例なので、他のものでもOK)。

■ Windows TeraTerm

■ Android JuiceSSH

■ iOS Terminus

※port number は 22

IP version は auto または IPv4

■ Mac OS X, Linux ssh コマンド (オプションとして login username を指定)

■ ssh -l j204xx cs.jga.kisarazu.ac.jp ← -l は、ハイフン エル (この記号を発音するときには、オプション エル という)

■ パスワードは学科演習マシン (3F, 4F) と同じ。

■ 演習用サーバ jes に接続

■ cs 上で ssh コマンドを使う。

■ ssh -l j204xx jes ← -l については上記を参照。

■ hostname ← 現在のホストが jes であることを確認。

■ exit ← これで jes からログアウト。

■ exit ← これで cs からログアウト。

今回のミッション: 演習の準備

6

- いろいろな場面でCtrl キーを使うので、Ctrl を入力できるようにしておくこと
 - 物理キーボードを使う。
 - Android 「日本語フルキーボード」というアプリを使う。
- FYI : jes について
 - cs からのみアクセスできる。
 - 基本的に、jes からは直接 the internet へ出ていけない。
 - 学内・学科内の いくつものネットワークからも独立している。
 - 専攻科を含めた全学年での使用を想定。
 - 学科サーバ上のhome directory の内容は、コピーしていない。
 - **重要**: プログラミングの演習は、jes にて行うこと。
cs 上では行わないこと。

今回のミッション

7

■ 基本的注意

- jes では、emacs は & をつけずに普通に起動すること。
 - & を付けてバックグラウンドジョブとして起動してしまった場合は、fg コマンドを用いてフォアグラウンドジョブにする。
- 演習が終わるまで、emacs を終了しないこと。

ファイルは C-x C-f で呼び出す。

 - C-z でサスペンドして、make コマンドや作成したプログラムを実行する。
 - fg コマンドで一時停止しているemacs を呼び出す。
 - 万一、プログラムを破壊してしまった場合、emacs のバッファに残っている可能性がある。何か編集(たとえばスペース、バックスペースを入力)して保存すれば復活できる。
- emacs を複数起動するのは不可。
 - emacs を起動する前に、jobs コマンドで既に起動していないかどうかを確認すること。

```
$ jobs
```

```
[1] + suspended emacs
```

```
$ fg %1
```

 ←この「1」は、[] 中のジョブ番号。

ジョブが1つの場合は fg だけでもOK。