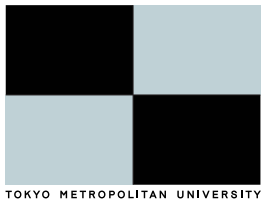
首都大学東京

TOKYO METROPOLITAN UNIVERSITY

情報通信応用実験
ネットワークプログラミング
チュートリアル

酒井 和哉



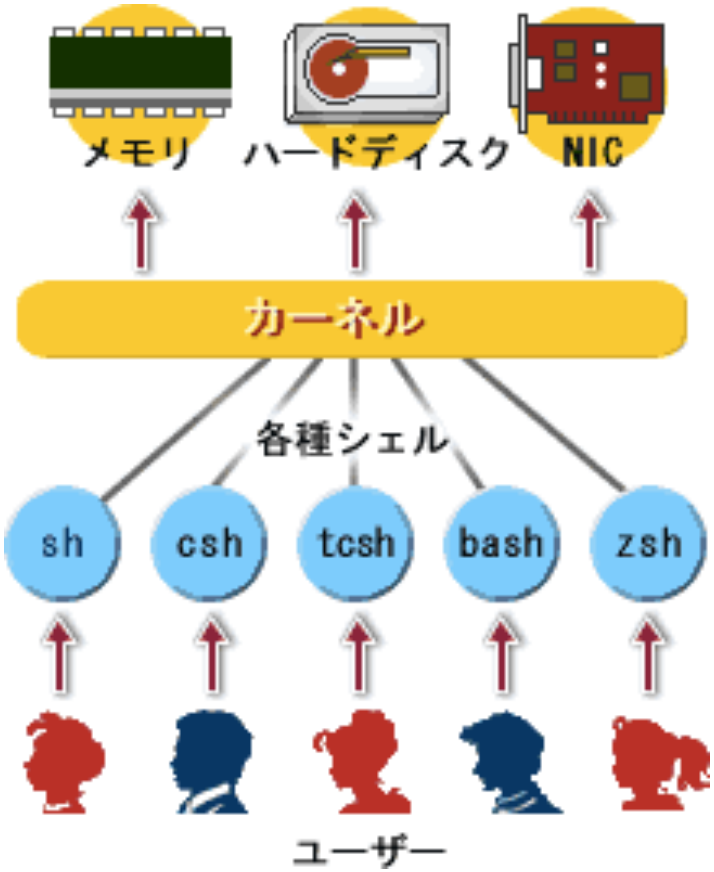
概要

1. シェル
2. Makefile
3. システムコール（ファイル入出力）
4. ソケット・プログラミング
5. デバッグ
6. 符号化・復号化
7. シグナルとコールバック

TOKYO METROPOLITAN UNIVERSITY

1. シェル

- シェル
  - カーネルを操作するためのインターフェース
  - 例）
    - Sh, csh, tcsh, bash, zsh
- カーネル
  - OSの中核、資源管理



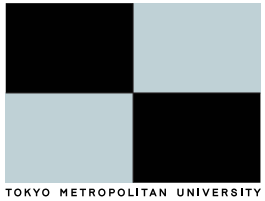
TOKYO METROPOLITAN UNIVERSITY

1．シェル（続き）

- ターミナルを起動して、コマンドを打つ

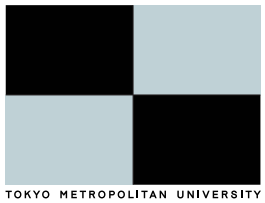
表．UNIXコマンドの例

ls	List service : カレントディレクトリのファイルとフォルダを表示
pwd	Present working directory : 現在のディレクトリを表示
cd	Change directory : ディレクトリを移動
mkdir	Make directory : ディレクトリを作成
rm	Remove : ファイル又はディレクトリを削除
cp	Copy : ファイル又はディレクトリのコピー
mv	Move : ファイル又はディレクトリの移動



1．シェル（続き）

- ネットワークコマンド
  - ifconfig, netstat, traceroute, etc.
- シェルスクリプト
  - コマンドを列挙したファイル
- 例）bashを用いたスクリプト
  - ex_shell.tar.gz



2. Makefile

- Makefileとは？
  - 1）コマンド、2）ファイル、3）ファイルの依存関係
  - “make” と入力すれば、自動的にコマンドを実行
  - スクリプトとの違い
    - 依存関係に基づいて更新が必要なファイルだけコンパイル
- サンプルソースコード
  - “sys_calls.tar.gz” の “Makefile”

TOKYO METROPOLITAN UNIVERSITY

2. Makefile（続き）

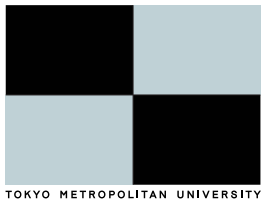
- “make”と入力
  - “all:”で指示したコマンドが実行される
- “make clean”と入力
  - “clean:” で指定したコマンドが実行される

```
CC   = gcc
SRC = code

all:
	$(CC) $(SRC).c -o SRC

clean:
	rm $(SRC)
```

gcc code.c -o code が実行される

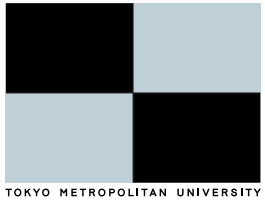
rm code が実行される

図. Makefileの例



3. システムコール

- システムコールとは？
  - オペレーティングシステムを直接操作する関数
- 参考ソースコード
  - “sys_calls.tar.gz” 内の“ex_io.c” と “ex_lseek.c”
  - “ex_lseek.c” は課題3で利用するので、自分で確認する
- 今回は、ファイル入出力を例とする
  - open(.)　　:ファイルを開く
  - read(.)　　:ファイルからデータを読み込む
  - write(.)　　:ファイルへデータを書き込む
  - オプション（自分で調べる）
    - O_RDONLY、O_WDONLY等、（読み込みモード、書き込みモード）



3. システムコール（続き）

- ファイル記述子（File descriptor）
  - 入力元や出力先を識別するための識別子（int型）

```
int fd;
char file_name[8] = “file.txt”
char buff[16];

fd = open(file_name, O_RDONLY);
read(fd, buff, 8)
```

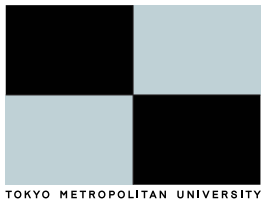
file.txtの内容：
Tokyo Metropolitan University

“buff”にコピーされる文字列
Tokyo Met

```
int fd;
char file_name[8] = “file.txt”
char buff[16] = “6-6 Asahigaoka”;

fd = open(file_name, O_WDONLY);
write(fd, buff, 8)
```

“buff”の内容：
6-6 Asahigaoka

“file.txt”に書き込まれる文字列
6-6 Asahi



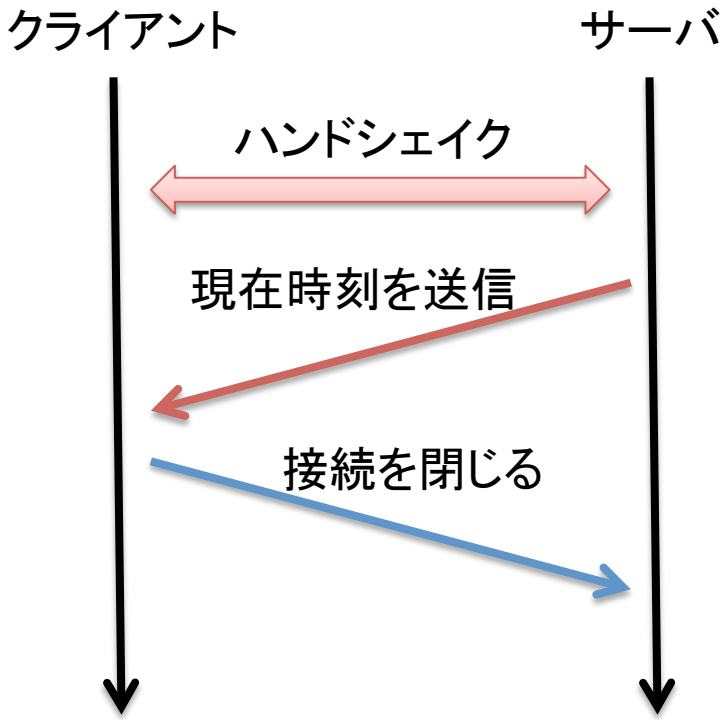
4. ソケットプログラミング

- ソケット
  - TCP/IP通信を行うためのインターフェース
  - ソケットにデータを入れたら、相手先のソケットからデータが出てくる
- トランスポートプロトコル
  - TCP　:コネクション指向型通信
    - TCPハンドシェイクして、ソケットを接続してから通信
  - UDP　:コネクションレス通信
    - プログラマから見ると、ほとんどIPとやりとり
- サンプルソースコード
  - “daytime.tar.gz”内のファイル

TOKYO METROPOLITAN UNIVERSITY

4. ソケットプログラミング（続き）

- TCPを基にしたDaytimeプログラム
  - サーバはポートを開き、クライアントからの接続を待機
  - クライアントがサーバに接続すると、サーバは現在時刻を返信

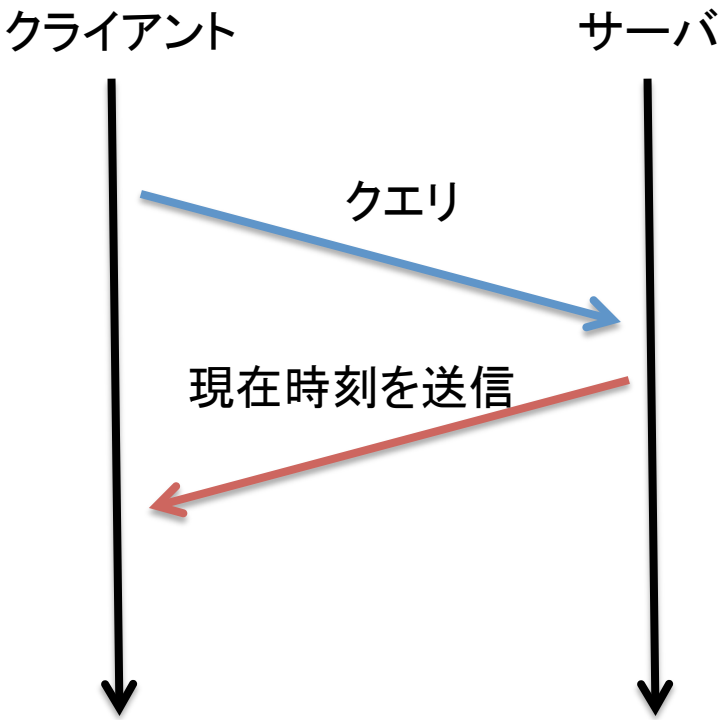


ソースコード
tcp_server.c と tcp_client.c

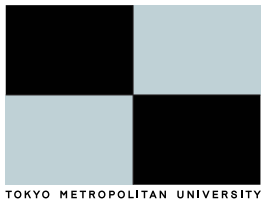
TOKYO METROPOLITAN UNIVERSITY

4. ソケットプログラミング（続き）

- UDPを基にしたDaytimeプログラム
  - サーバはポートを開き、クライアントからのクエリを待機
  - クライアントは、クエリ（単なる文字列）をサーバへ送信
  - クライアントからクエリを受信すると、サーバは現在時刻を返信



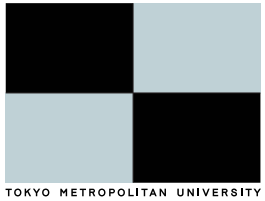
ソースコード
udp_server.c と udp_client.c



5. デバッグ

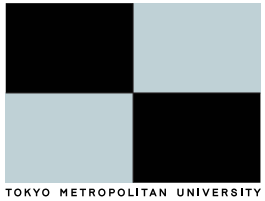
- デバッグとは？
  - プログラムのバグをとること
  - 例）segmentation fault
- ツール
  - gdb
- サンプルソースコード
  - ex_gdb.tar.gz

```
# コンパイル時のオプション
-g    : デバッグ情報追加
-O0  : コードの最適化を行わない

# デバッガの起動
$ gdb program
(gdb) run [argv]

# デバッガコマンド
where
print var
```



６．符号化・復号化

- 符号化と復号化
  - 整数型や文字列をバイトへ変換
  - バイトを整数型や文字列に変換
- 例
  - コマンド、uint8_t cmd（１バイト）
    - upload 0x1, download 0x2, ack 0x3
  - ファイル名、char file_name[16]（１６バイト）
  - ファイルサイズ、int file_size（４バイト）
- サンプルソースコード
  - “codec.tar.gz”、課題3.2で必要なので、各自で勉強する



7. シグナルとコールバック

- シグナルとは？
  - 割り込み、タイマー処理
- コールバックとは？
  - タイマーが切れた時に、関数を呼び出すこと
- 使用例
  - ユーザからのパスワード入力を待機、10秒以内に何も入力されなければ、処理を打ち切る
  - クライアントがサーバへパケットを送信し、ACKを待機、5秒以内にACKが届かなければ、パケットを再送する
- サンプルソースコード
  - “sys_calls.tar.gz” 内の “ex_signal.c”
  - 課題4で必要なので、各自で勉強する