情報処理論1a

2章「X/プロセス/システム」

中島康彦

§2. 1 X Window System概要

X Window System

- 当初MITが開発したウィンドウシステム
- ネットワーク指向，様々なプラットホーム上で動作
- 現在では事実上の業界標準

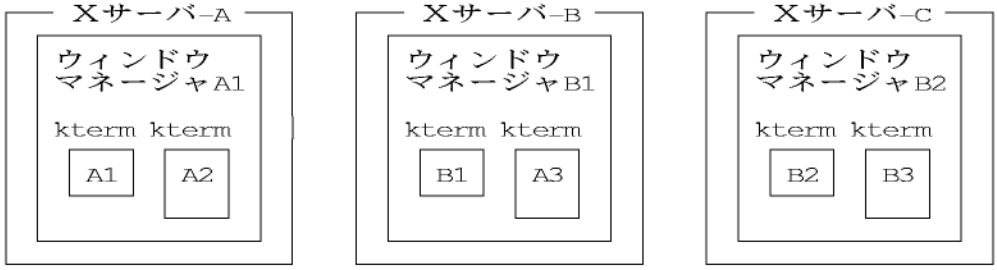
- Xサーバ … 計算機に固有
  画面/キーボード/マウスを制御

- Xクライアント … 個々のプログラム
  ウィンドウマネージャ(fvwm/twm等)
  メニュー/タイトルバーの制御も1クライアント
  個々のアプリケーション(kterm/xclock等)
  遠隔地の異機種計算機のプログラムでも表示可能

§2. 1 X Window System概要(続き)

本質的にネットワーク指向であるということ

- 画面の状況(外観は3台とも同じ)



- 実際の動作状況(非力な計算機には軽い仕事)

§2. 1 X Window System概要(続き)

xクライアントとして動作するビデオキャプチャカードがあれば，遠隔地から閲覧可能.

- 電波が直接届かないTV放送
- 防犯カメラ
- TV電話

遠隔地のファイルも違和感なく操作可能
ファイアウォールを越えることも知識があれば容易

- お金を払えば消費者としてサービスを手に入れることは可能
- 知恵で解決するか，お金で解決するかのトレードオフ

X Window Systemの説明はここまで.

- 以下では，プログラム一般の話としてxクライアントを用いる

§2. 2 様々なプログラム

終了するプログラム(実行時に&が不要)

- `bggen 0 0 0 0 0 255 255 255 255 >back`
- `xv -root -max -quit back`
  背景の生成，設定

- `xwd >back`（続けて適当なウィンドウを左クリック）
- `xwdtopnm back | xv -root -quit -`
  画像の取り込み，背景の設定

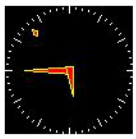
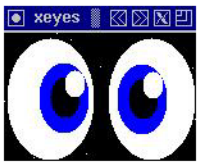
xクライアントではない他のコマンド例

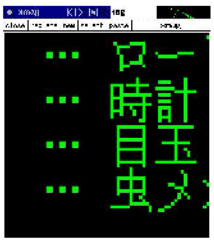
- `date`
- `cal`
- `cal 2005`
- `cal 4 2005`
- `echo 123456 | rev`

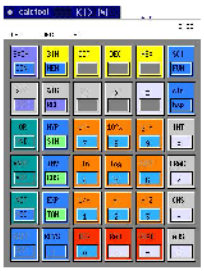
§2. 2 様々なプログラム(続き)

動き続けるプログラム(実行時に&が必要)

- `xclock -fg white -bg black -hd red -hl yellow -update 1 -padding 0 &`
- `xeyes -fg blue &`
- `xmag &`
- `calctool &`
- `xload -fg blue -bg yellow -update 3 &`
  ロードアベレージ(走行状態プロセス数の平均)






動作中のXクライアントの一覧表示

- `xlsclients`

§2. 3 プロセス

プログラム … 命令の静的な集まり

プロセス … プログラムが走行する動的な状態

- **ps -ef**
  システム内の全プロセスを表示

利用コード	プロセスID	開始時刻	関連ターミナル	実行時間	コマンド

```
     UID   PID  PPID  C    STIME TTY          TIME   COMMAND
ynakashi  8477  8475  1 18:52:27 pts/0        0:00   -tcsh
ynakashi  8504  8477  7 18:52:56 pts/0        0:00   ps -ef
e00x0000   788     1  0  1月 29  ?            2:38   perl test.pl
```

- **ps -f**
  現ターミナル(**kterm**)に関連するプロセスのみ
- **kill** プロセス番号
  該当プロセスを強制終了

§2. 3 ジョブ

ジョブ … ひとまとまりの仕事

1つまたは複数のプロセスから構成される.

- フォアグラウンドジョブ … キー入力を受け付ける現ジョブ
  &を付けない場合に相当
- バックグラウンドジョブ … キー入力が切り離されたジョブ
  &を付ける場合に相当

- **jobs**
  現シェル(**tcsh**)に関連するバックグラウンドジョブを表示

ジョブ番号	状態	ジョブ

```
[1] + Running xclock -fg white -bg black -hd red -hl yellow ..
[2] - Running xeyes -fg blue
[3]   Running xmag
[4]   Running calctool
[5]   Running xload -fg blue -bg yellow -update 3
```

§2. 3 ジョブ制御

- [Ctrl]+Cを押すと
  現ジョブ(フォアグラウンドジョブ)が強制終了

1. ktermを起動
   ウィンドウの無い場所で左ドラッグしktermを選択

2. calctoolを起動(フォアグラウンドジョブ)
   % calctool

3. [Ctrl]+Cを入力しcalctoolを強制終了
   ^C
   %

§2. 3 ジョブ制御(続き)

- [Ctrl]+Zを押すと
  現ジョブがバックグラウンドへ遷移し停止

1. calctoolを起動(フォアグラウンドジョブ)
   % calctool

2. [Ctrl]+Zを入力しcalctoolをバックグラウンドへ
   ^Z
   [1] + Suspended calctool

3. ジョブ状態を表示
   % jobs
   [1] + Suspended calctool

§2. 3 ジョブ制御(続き)

- bg %job番号
  バックグラウンドジョブの実行再開

1. ジョブ[1]をバックグラウンドで実行再開
   ```
   % bg %1
   [1] calctool &
   ```

2. [Ctrl]+Cを入力しても何も起きない
   ```
   % ^C
   ```

3. ジョブ状態を表示
   ```
   % jobs
   [1] + Running calctool
   ```

§2. 3 ジョブ制御(続き)

- fg %job番号
  フォアグラウンドへ遷移し実行再開

1. ジョブ[1]をフォアグラウンドで実行再開
   ```
   % fg %1
   calctool
   ```

2. [Ctrl]+Cを入力しcalctoolを強制終了
   ```
   ^C
   %
   ```

§2. 3 ジョブ制御(続き)

- kill %job番号
  該当ジョブを強制終了

1. calctoolを起動(バックグラウンドジョブ)
   % calctool &
   [1] xxxxx

2. ジョブ状態を表示
   % jobs
   [1] + Running calctool

3. killコマンドによりcalctoolを強制終了
   % kill %1
   [1] Terminated calctool

§2. 3 ジョブ制御(続き)

- wait
  Running中の全バックグラウンドジョブを待つ

1. calctoolを起動(バックグラウンドジョブ)
   % calctool &
   [1] xxxxx
2. ジョブ状態を表示
   % jobs
   [1] + Running calctool
3. waitコマンドによりcalctoolの終了を待つ
   % wait
4. calctoolのQUITボタンを押すとwaitが終了する
   [1] Done calctool

§2. 3 ジョブ制御(続き)

何でこんなことを知る必要があるでしょう?

- ネットワーク上のサービスを制御しようと思ったら, プロセスが制御できなければなりません.
- お金を払って誰かに作ってもらえば, マウスでカッコ良くできます. しかし, できることは所詮同じ.
- 極端な言い方をすると, IT化のコスト=見栄えのコスト
- 見栄えのためのエネルギー消費は馬鹿にならない.

§2. 4 システムの状態

- `hostname` ⇒ ホスト名の表示
- `last` ⇒ ログイン履歴の表示
- `w` ⇒ ユーザ名, ログイン時刻, 実行中コマンド
- `finger` ⇒ ユーザ情報(名前など)
- `df` ⇒ ディスクの空き状況

DISK名	総容量	使用中	空容量	%	ディレクトリ名
Filesystem	kbytes	used	avail	%used	Mounted on
/dev/vg00/lvol3	107669	20458	76444	21%	/
/dev/vg00/lvol1	59797	14238	39579	26%	/stand
/dev/vg00/lvol7	95973	57200	29175	66%	/var
/dev/vg00/lvol6	490645	388685	52895	88%	/usr
/dev/vg00/lvol5	122821	3	110535	0%	/tmp
/dev/vg00/lvol4	920931	431496	397341	52%	/opt
us201:/usr/local	1025617	539955	383100	58%	/tmp_mnt/usr/local
nfs0:/h0	133426784	20634956	99449149	17%	/tmp_mnt/home0
nfs1:/h1	133426784	12809922	107274183	11%	/tmp_mnt/home1

↑
下から2行が全ホームディレクトリに関する情報

§2. 4 システムの状態(続き)

- `uptime` ⇒ 起動時刻, ユーザ数, システム負荷など
- `ruptime` ⇒ 他のホストも含む`uptime`

```
ホスト名      経過時間    ユーザ数        ロードアベレージ
us201     up  39+04:37,  0 users,  load 0.08, 0.06, 0.07
uw122     down     ??:??
uw201     up       13:38,  0 users,  load 1.01, 1.05, 1.04
uw202     up       13:38,  1 user,   load 1.06, 1.02, 1.02
uw203     up       13:38,  0 users,  load 1.00, 1.00, 1.01
```

- `who` ⇒ ユーザ情報(ログイン時刻など)
- `rwho -a` ⇒ 他のホストも含む`who`

```
利用コード  ホスト名          ログイン時刻      IDLE
s00t2040    uw208:ttyp5   Jan 31 17:32   3:08
t26t0306    uw208:ttyp2   Jan 31 16:39   3:44
t25u0649    uw204:ttysa   Jan 31 13:41   5:46
t25u0649    uw204:ttysb   Jan 31 12:53   5:36
ynakashi    uw209:0       Jan 31 20:41
ynakashi    uw211:0       Jan 31 20:41    :01
```

ただし個人利用の場合は利用価値はない

§2. 5 他システムへのログイン

昔ながらの方法 (と言っても20年前)

- `rlogin` ホスト名
  現ユーザIDによる他ホストへのログイン
  パスワードの入力のみ(設定により不要にできる)
  1. `rlogin`コマンドを入力
     % **rlogin xxx** (`ruptime`に表示されたホスト名)
  2. 他ホストからログアウト
     % **logout** (`exit`でも良い)

- `telnet` ホスト名
  必ずユーザIDとパスワードの入力が必要
  1. `telnet`コマンドを入力
     % **telnet xxx** (`ruptime`に表示されたホスト名)
  2. 他ホストからログアウト
     % **logout** (`exit`でも良い)

§2. 5 他システムへのログイン(つづき)

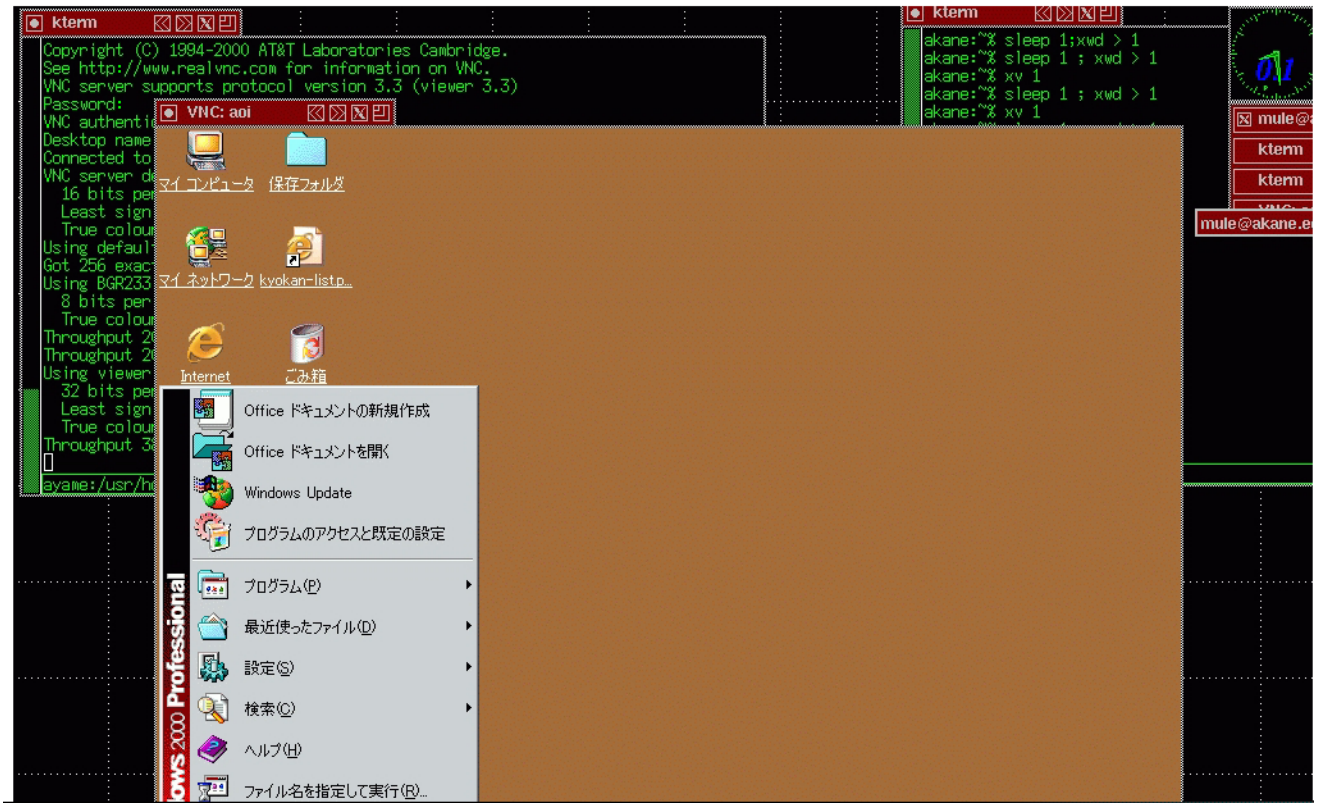
ここ数年はセキュリティ強化のために，通信内容が暗号化されるssh （Secured Shell） への移行が進んでいる.

- ssh ホスト名
  現ユーザIDによる他ホストへのログイン
  パスワードの入力のみ(設定により不要にできる)
  1． sshコマンドを入力
  % ssh xxx （ruptimeに表示されたホスト名）
  2． 他ホストからログアウト
  % logout （exitでも良い）
  トンネリングによりファイアウォールを越えることもできる

§2. 6 遠隔コンピュータの利用

ファイアウォール内遠隔コンピュータの利用

FW内A ⇒ GW-B ⇒ GW-C ⇒ FW内D



§2. 6 遠隔コンピュータの利用(つづき)

ネットワーク上に，管理できるサーバがあると，様々な基本サービスを展開できる.

- 通信拠点　　（smtp,pop,fax,TV電話）
- ファイルサーバ(ftp,nfs,smb,webdab)
- WEBサーバ　（http,proxy）
- データベース　（pgsl,mysql）
- 計算サーバ　（ssh）

§2. 7 例題

w の実行結果をメールせよ.

- 以前のシステムは共用だったので他人のログイン状況が表示されたが，現システムはVMwareなので，自分に関する情報しか表示されない.

§2. 8 今日の課題

ps -ef を実行し，自分のユーザIDにて動作しているプロセスを全て調べて，それぞれ何であるかをレポートせよ．言うまでもなく，ユーザごとに実行結果は異なる．

宛先: nakashim@econ.kyoto-u.ac.jp
件名: unix1-学生番号
---
kterm … ターミナル
<など>
名前(忘れずに)

- kterm上で，コピーしたい範囲を左ドラッグで選択し，webmail上の本文位置で中クリックすることにより，ktermからwebmailへ文字をコピーすることができる．

今日はここまで