X Window System 入門 (GTK に入る前に)

● X Window System とは

Unix/Linux で利用されるグラフィカルユーザインターフェース (GUI) 環境．ほとんどの Unix 系 OS に標準で搭載されており，他の OS にも移植されている。クライアントがサーバの機能を呼び出して使う分散構造になっており，アプリケーションソフトや OS の処理はクライアントが，画面表示や入出力はサーバが行う．

X Window 上のデスクトップ環境の代表的なものに，GNOME や KDE などがある．外観や背景など設定や，マウスによる操作インターフェイスを提供していて，コアアプリケーションであるウィンドウマネージャは，ウィンドウに関する見た目や操作を提供している．

演習で用いる予定の GTK+ライブラリと X Window の関係は次のようになる．

アプリケーション・プログラム
デスクトップ環境（背景，ウィンドウマネージャ)
GTK+/GDK (ツールキットレベル・ライブラリ)
XLib (X 最下層ライブラリ)

[1] 特徴

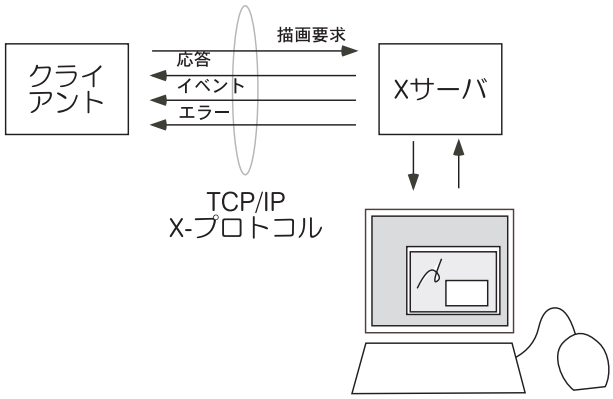
- ネットワーク指向 – プログラムの表示 (絵，テキスト,…) を他の計算機に送れる．便利.
- アプリケーション・プログラムはデバイス独立，OS 独立，ベンダ独立 – Unix/Linux, Windows, Mac でも使える．ただし，マシン毎の X サーバ (後述) が必要.
- Unix/Linux の世界での GUI 構築における基礎技術

[2] クライアント・サーバ・モデル

X Window System では，画面表示とキーやマウスなどの入力を担当するサーバと，それを利用して動作するクライアント（アプリケーション）とい

うクライアント／サーバ構成になっている．X サーバは，このサーバ機能を指す．X 端末のように専用ハードウェアになっていて，クライアントとは別マシンの場合もあれば，一般的なワークステーションのように単なるソフトウェアの場合もある．一般のサーバクライアントモデルと比べて，ユーザーサイドのプログラムがサーバになっている点に注意．

PC Unix/Linux では，Xorg がよく使われている．



[3] X プロトコル

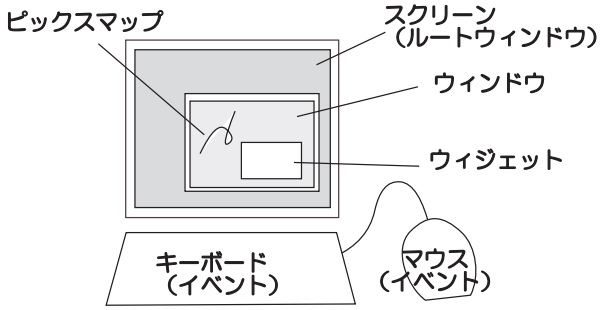
- リクエスト – ウィンドウの生成、操作、図形出力等のクライアントからサーバへの発行、または各種情報の要求。
- リプライ – 必要な情報をクライアントへ返すためにサーバからクライアントに送るメッセージ。
- イベント – サーバ側で非同期に発生する事象をクライアント側に知らせるメッセージ。
- エラー – エラーの発生を検知されたことをクライアントに知らせるメッセージ。

[4] X 入出力装置

ディスプレイ キーボードやマウス (などのポインティングデバイス)、一つ以上のモニタ等により構成される WS

スクリーン 表示するモニタ．2 次元座標で位置が指定される画素の集まり．特定の計算機上のメモリ (フレームバッファ) が画素と対応する．

ウィンドウ Xアプリケーションとして表示されるスクリーン上の区画．スクリーンとは独立な座標系を持つ．



● X プログラミング・モデル

- Display (構造体) – ディスプレイの情報を持つ構造体．プログラムからこの構造体を操作することで，ディスプレイを扱う．環境変数 `DISPLAY` がデフォルトのディスプレイを示す．
- Window, Subwindow, ... – ウィンドウやウィンドウ内のウィンドウ．これらも構造体．親子関係，2 次元座標．
- Pixmap – 描画可能な構造体．メモリ上に絵を描くというイメージ．Pixmap を Window に貼ると絵が出てくる．
- Cursor – マウスカーソル (矢印), テキストカーソル (キー入力位置を示す印)
- Font – 文字のグラフィカルなパターン．いろいろな書体がある．かなや漢字用のフォントを使わないと，日本語が表示できない．
- GC (Graphic Context) – さまざまな グラフィック要素 (描画されるドット，線，文字，画像 (タイルや塗りつぶし) など) に必要な各種情報 (リソース)．線種や前景色などウィンドウ内に描画されるものに適応される．
- Window マネージャ – ルートウィンドウやウィンドウの (共通の) 概観などを扱うプログラム．アプリケーションにいろいろ影響を与える．
- イベント駆動型プログラム

GUI を持つプログラムは，ボタンをクリックしたり，テキストを入力したりなどの，イベントによってプログラムの流れが決定される．これをイベント駆動 (イベントドリブン) という．

X でのイベントには，ポインタに関連したイベント (ButtonPress，ButtonRelease，MotionNotify，[EnterNotify，LeaveNotify)，キーボード関連イベント (KeyPress，KeyRelease，FocusIn，FocusOut，KeymapNotify，MappingNotify) などがある．

● X アプリケーションをイメージしてみよう

- 端末プログラムがやっている事
- `firefox` ブラウザのやっている事
- ウィンドマネージャがやっている事

参考文献

[1] 「プログラミング X Window」，Naba Barkakati 著，ラジオ技術社

[2] 「Xlib プログラミング・マニュアル 1」，Adrian Nye 著，ソフトバンク

[3] 「X-Window Ver.11 プログラミング【第 2 版】」，木下凌一・林秀幸著，日刊工業新聞社

● X サンプルプログラム – マウスで絵を描く

```
/*
 * マウスにより絵を描くプログラム part 2 (mdraw2.c)
 *
 *    描画領域に Pixmap を使用
 *    cc -o mdraw mdraw2.c -I/usr/X11R6/include -L/usr/X11R6/lib -lX11
 *
 */
#include <X11/Xlib.h>
#include <X11/Xutil.h>
#include <stdio.h>

#define  WIN_WIDTH   500
#define  WIN_HEIGHT  400

main()
{
    Display        *dsp;
    Window         wd;
    GC             gc;
    Pixmap         pmap;
    XEvent         ev;
    int            xs, ys, xe, ye;
    unsigned long  black,white;

    dsp = XOpenDisplay(NULL);
    black = BlackPixel(dsp, 0);
    white = WhitePixel(dsp, 0);
    wd = XCreateSimpleWindow(dsp, RootWindow(dsp, 0), 350, 50,
                             WIN_WIDTH, WIN_HEIGHT, 2, white, black);

    pmap = XCreatePixmap(dsp, wd, WIN_WIDTH, WIN_HEIGHT,
                         DefaultDepth(dsp, 0)); /* ピクスマップの
生成 */
    gc = XCreateGC(dsp, wd, 0, 0);
    XSetLineAttributes(dsp, gc, 5, LineSolid, CapRound, JoinRound);
    XMapWindow(dsp, wd);

    XSetForeground(dsp, gc, black);     /* ピクスマップ領域 */
```

```
    XFillRectangle(dsp, pmap, gc, 0, 0, /* の背景を黒で初期化*/
                   WIN_WIDTH, WIN_HEIGHT);
    XSetForeground(dsp, gc, white);     /* 描画色を白に設定 */

    XSelectInput(dsp, wd, ButtonPressMask | ButtonMotionMask |ExposureMask);

    while(1) {
        XNextEvent(dsp, &ev);
        switch(ev.type) {
            case Expose:       XCopyArea(dsp, pmap, wd, gc, 0, 0,
                                         WIN_WIDTH, WIN_HEIGHT, 0, 0);
            case ButtonPress:  xs = ev.xbutton.x;
                               ys = ev.xbutton.y;
                               break;
            case MotionNotify: xe = ev.xbutton.x;
                               ye = ev.xbutton.y;
                               XDrawLine(dsp, pmap, gc, xs, ys, xe, ye);
                               XCopyArea(dsp, pmap, wd, gc, 0, 0,
                                         WIN_WIDTH, WIN_HEIGHT, 0, 0);
                               xs = xe;
                               ys = ye;
        }
    }
}
```