

LPICスタートアップ無料セミナー@大阪

X Window System
スケジュール管理(cron,at)、ログ管理(syslog)
xinetdとTCP Wrappers

2007年2月23日



担当講師

パナソニック ITカレッジ ビジカル

寺井 健一郎

<http://panasonic.jp/busicul>



Key Points

- X Window System
 - Xはクライアント・サーバー形式で動作していること
 - Xサーバーの設定ファイルの内容と起動方法
 - ウィンドウマネージャ、ディスプレイマネージャの役割
- スケジュール管理・ログ管理
 - cronを利用した定期的なジョブの実行方法
 - atを利用した一時的なジョブの実行方法
 - システムログの一元管理のしくみ、ログファイルの監視・管理方法
- xinetdとTCP Wrappers
 - サーバプロセスの起動方法
 - xinetdによるクライアントのアクセス制限
 - hosts.allow、hosts.denyファイルによるクライアントのアクセス制限

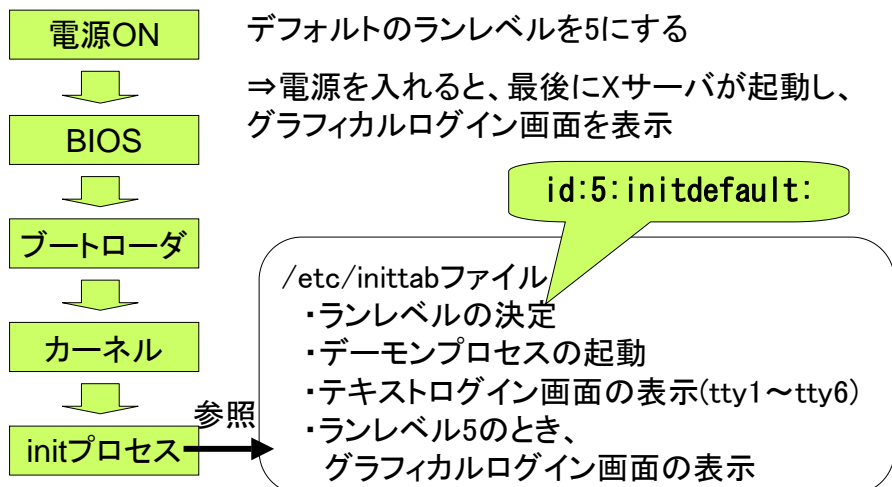
X Window System

X Windowの概要

- UNIX上で動くGUI環境として、MIT(マサチューセッツ工科大学)で開発
 - 前身は W という名前
 - X11R7(バージョン11 リリース7)が最新
- LinuxにおけるX Window System
 - XFree86 ProjectがフリーのX Window Systemを開発
 - ライセンスの問題があり、現在はX.Org Foundationに移管
- LPIC101試験の新範囲で・・・
 - X Windowのトピック(Topic 110)に、X.Orgが追加される

X Window System

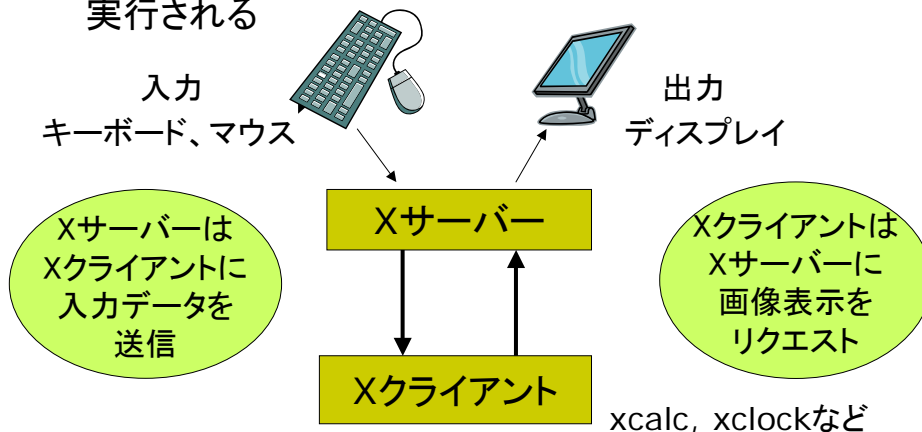
ランレベルとX Window



X Window System

X Windowのしくみ

- 通常、1台のホスト内でクライアント・サーバー形式で実行される



X Window System

Xサーバーの設定ファイル

- Xサーバーの設定ファイル
 - XFree86の場合 XF86Config (/etc/X11/XF86Config)
 - Xorgの場合 xorg.conf (/etc/X11/xorg.conf)
- “セクション”に分かれている
- 主なセクション
 - Files 使用するフォントと色
 - InputDevice キーボードとマウスの設定
 - Monitor モニタの設定
 - Device ビデオカードの設定
 - Screen どのモニタがどのビデオカードにつながっているか？ 解像度はココで指定

X Window System

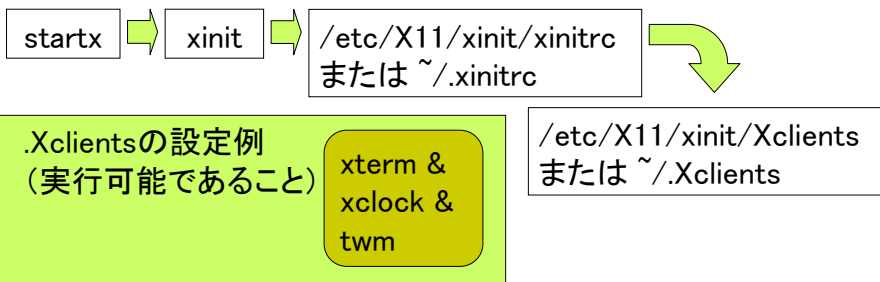
Xサーバーの起動①

- ▶ ランレベルが3の状態で行います(テキストログイン画面)
- ログイン後、**xinit** と入力
 - Xサーバーとxtermだけが稼働している状態
 - xtermから、Xアプリケーションやウィンドウマネージャを起動
 - xtermを終了させるとテキストログイン画面に戻る
- ウィンドウマネージャ
 - ウィンドウの外観・操作性を決定
 - 枠をつける、最大化最小化、移動 など
 - twm、mwm、metacity、kwm などから、自分の好みのものを選ぶ

X Window System

Xサーバーの起動②

- ログイン後、**startx** と入力
 - Xサーバー、ウィンドウマネージャ、Xアプリケーションが自動的に起動する
 - ウィンドウマネージャ、Xアプリケーションをユーザーが指定したい場合、ホームディレクトリに **.Xclients** ファイルを作成



X Window System

ディスプレイマネージャ

- xinit, startxによる起動
 - ログイン→Xサーバー起動→Xアプリケーション起動 の順

- ディスプレイマネージャ
 - Xサーバーを起動後、ログイン画面を表示し、ユーザー認証を行う (XDMCPプロトコル)
 - XDM, GDM, KDM などの種類がある

- テキストログイン画面から xdm と入力
 - ログイン画面からユーザー名とパスワードを入力
 - Xアプリケーションが自動起動する

X Window System

ディスプレイマネージャ

- ログイン後に自動起動させたいアプリケーションの指定方法
 - /etc/X11/xdm/Xsession (システム共通)
 - ホームディレクトリの .xsession (ユーザー独自)
 - .xsessionがないときは、.Xclientsを参照

.xsessionの設定例
(実行可能であること)

```
xterm &  
xcalc &  
mwm
```

X Window System

Xアプリケーション

- Xサーバーと接続して稼動するプログラム
 - xterm(ターミナル)、xclock(時計)、xcalc(電卓)など
- Xアプリケーション起動時の属性
 - 前面色、背面色、フォント、サイズ、表示位置など
 - /usr/lib/X11/app-defaults ディレクトリのファイル
 - ユーザー独自の設定は、ホームディレクトリの .Xresources ファイルで

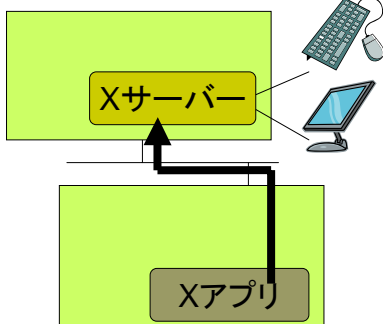
.Xresourcesファイルの
設定例

```
xterm*background: black
xterm*foreground: yellow
xclock*geometry: 100x150+100+200
```

X Window System

ネットワーク経由でのXの利用

サーバー: ホスト名 host02
IPアドレス 192.168.0.102



クライアント: ホスト名 host01
IPアドレス 192.168.0.101

- サーバー側
xhostコマンドでクライアントの
アクセスを許可
(例) xhost + host01

- クライアント側
DISPLAY環境変数を設定し
アプリケーションを実行
(例) DISPLAY=host02:0
export DISPLAY

ホスト名: ディスプレイ番号

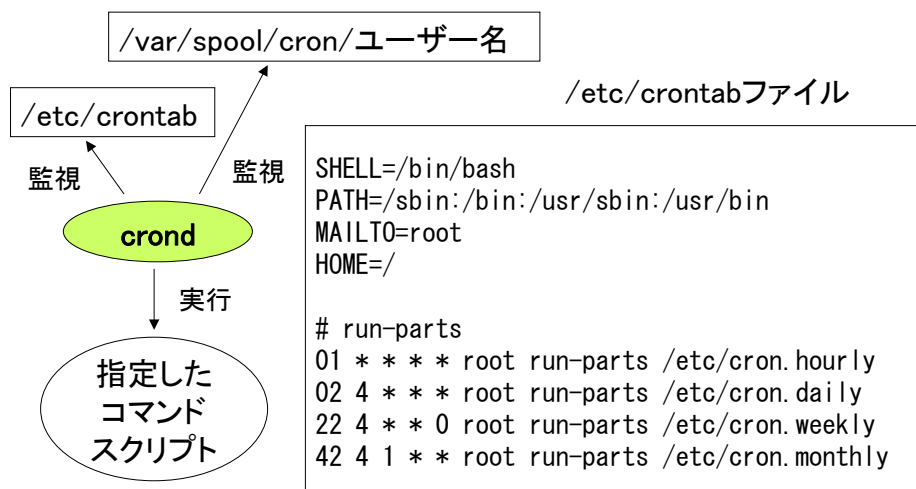
スケジュール管理

cronのしくみ

- cron(クローン) 定期的にジョブを実行する機能
 - 「毎日午前9時に実行」
 - 「毎週日曜日の午後3時に実行」
...など バッチジョブの実行
- システムのcron
 - Linuxインストール時に自動的に設定
 - ログのローテーション、/tmp内のファイルの削除 など
 - /etc/crontabファイルに基づき、/etc/cron.hourly、
/etc/cron.dailyなどのディレクトリにあるスクリプトを実行
- ユーザー指定のcron
 - /var/spool/cron/ディレクトリにあるユーザー名のファイル
でスケジュールを管理

スケジュール管理

cronのしくみ



スケジュール管理 cronの設定

□ 設定フォーマット

* * * * * (ユーザー) コマンド 引数 ...
分 時 日 月 曜日 曜日は0(日曜)~6(土曜)であらわす

□ 設定例

0 9 * * * prog1 毎日午前9時にprog1を実行
0 9-17 * * * prog2 毎日9:00, 10:00, ..., 17:00にprog2を実行
30 5, 17 * * * prog3 毎日5:30, 17:30にprog3を実行
0 */2 * * * prog4 毎日2時間ごとに(0:00, 2:00, ...)prog4を実行

🌐 2分ごとに `date >> datefile` を実行するには、どのように登録すればよいですか？

スケジュール管理 cronの設定

□ ユーザーによるcronの指定

- 各ユーザーが `crontab` コマンドで登録
`crontab -e` 設定、`crontab -l` 表示、`crontab -r` 削除
`crontab -e` でviエディタが起動
スケジュール記述後、保存・終了

□ cronの実行ログ

- syslogd経由で、`/var/log/cron` ファイルに書き込まれる

□ anacron

- 指定した時刻にホストがダウンしていてcronが実行できなかった場合に、あとから実行してくれる機能

スケジュール管理

atのしくみ・設定

□ 日時を指定してジョブを実行

- cronと違い、指定日時に1回だけ実行
- 指定時刻にマシンが稼動していなかったら、ジョブは無視

/var/spool/at

監視

atd

実行

指定した
コマンド
スクリプト

□ 設定方法

- at 日時 と入力
 - at 15:00 今日の15:00
 - at noon + 3 days 3日後の正午
 - at 5pm + 2 weeks 2週間後の17:00
- 対話形式で、実行するコマンドを入力
- Ctrl+D で入力終了

スケジュール管理

atのスケジュール確認・削除

□ at関連コマンド

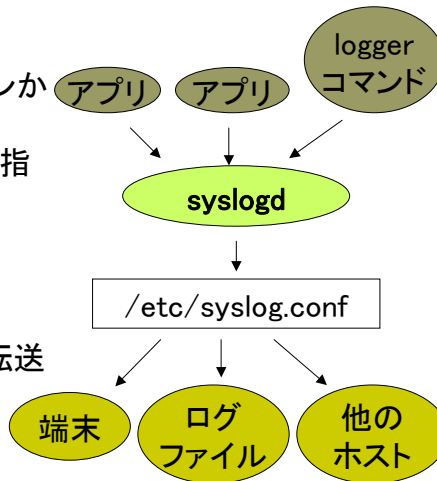
- 実行待ちジョブのスケジュール確認
 - atq または at -l
- ジョブの内容確認
 - at -c ジョブ番号
- ジョブの削除
 - atrm ジョブ番号 または at -d ジョブ番号

ログ管理

syslogのしくみ

□ システムログの一元管理

- syslogdが各アプリケーションからのログを収集
- /etc/syslog.confファイルの指定に従って処理
 - ファイルに書き出す (通常/var/logの下)
 - 画面に表示する
 - 他のホストのsyslogdに転送など



ログ管理

syslogの設定

□ /etc/syslog.conf ファイルの構文

ファシリティ. プライオリティ アクション
出力元. 重要度 出力先

ファシリティ

authpriv	ログイン認証関係
lpr	プリンタ
mail	メール
news	ニュースグループ
cron	cron
*	すべてのファシリティ

プライオリティ

emerg	重大な問題発生	高
alert	危険な状態	↑
crit	致命的	
err	一般的なエラー	↓
warning	警告	
notice	通知	低
info	情報	
*	すべてのプライオリティ	

ログ管理

syslogの設定

アクション

ファイル名	指定したファイルに出力
@ホスト名	指定したホストのsyslogdに転送
/dev/console	端末に出力
*	ログイン中の全ユーザーの端末

設定例

```
*. info:mail.none /var/log/messages
mail.* /var/log/maillog
cron.* /var/log/cron
*. emerg *
news.=crit /var/log/news/news.crit
news.=err /var/log/news/news.err
news.notice /var/log/news/news.notice
```

ログ管理

ログの監視・ローテーション

- loggerコマンド
 - syslogdにログを通知するコマンド
 - logger -p プライオリティ メッセージ文字列
- ログファイル
 - tail -f ファイル名 でリアルタイムにログを監視できる
(新しいメッセージが書き込まれるたび、末尾に追加)
- ログのローテーション
 - ログファイルのサイズが大きくなるのを防ぐ
 - /etc/logrotate.confで設定

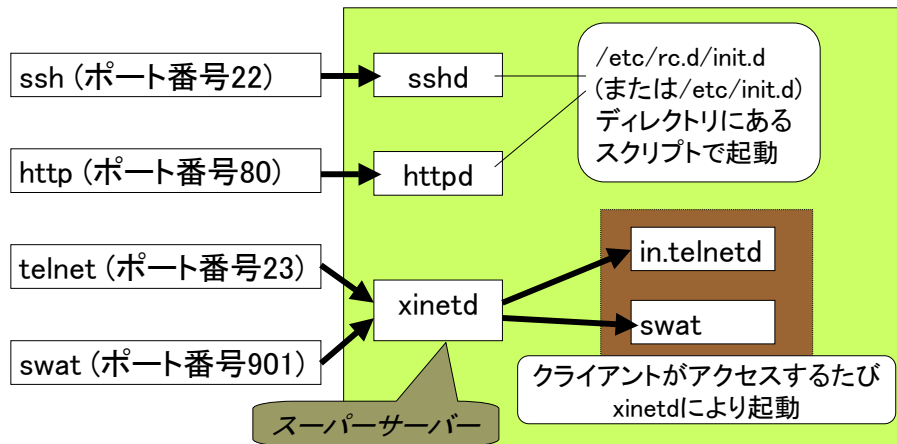


xinetd

デーモンプロセスの起動

クライアント

サーバー



xinetd

デーモンプロセスの起動

- デーモン(daemon : 守護神)プロセス
 - メモリ上に常駐しているプロセス
 - サーバーとして、クライアントの接続を待っている
 - プロセス名の末尾にdがつくことが多い

- スーパーサーバーが起動するプロセス
 - クライアントの接続はxinetd(またはinetd)が受付
 - 接続要求があれば起動され、接続切断で終了
 - 必要時だけ起動されるので、リソースの効率化が図れる
 - クライアントへの応答は遅くなる

xinetd

スーパーサーバーの設定

- xinetdからの呼び出しを許可する(拒否する)設定
 - chkconfigコマンドの利用
 - chkconfig サービス名 on
 - chkconfig サービス名 off
 - chkconfig --list サービス名 (確認)
 - 設定ファイルを直接変更
 - /etc/xinetd.conf (全体の設定)
 - /etc/xinetd.dディレクトリにある各サービスの設定ファイル

xinetd

スーパーサーバーの設定

- /etc/xinetd.d/telnetの設定例

```
service telnet
{
    disable = no           ←noで有効、yesで無効
    socket_type = stream  ←TCPプロトコルを使用して
    wait = no
    user = root           ←rootユーザーで
    server = /usr/sbin/in.telnetd
                          ↑このプログラムを起動
    log_on_failure += USERID
}
```

xinetd

スーパーサーバーの設定

- 追加可能なパラメータ
 - only_from サービスの利用を許可するホストを指定
 - no_access サービスの利用を拒否するホストを指定
 - access_times サービスの利用時間帯を指定

- パラメータ設定時の注意
 - = の両側にスペースが必要
 - 設定ファイルを直接修正した場合はxinetdのリロードが必要

- ログ
 - xinetd起動時のログ /var/log/messages
 - アクセス拒否のログ /var/log/secure

TCP Wrappers

TCP Wrappersのしくみ

- tcpd
 - inetd(xinetdの前身)で使われていたアクセス制御の仕組み
 - tcpdがクライアントのアクセス制御を行い、inetdがサーバーを呼び出す
 - スタンドアローン型でも、この機能が使えるものがある (vsftpd, sshdなど)

- 設定ファイル
 - /etc/hosts.allow 接続を許可するホストを指定
 - /etc/hosts.deny 接続を拒否するホストを指定
 - 書式はいずれも サービス名: ホスト

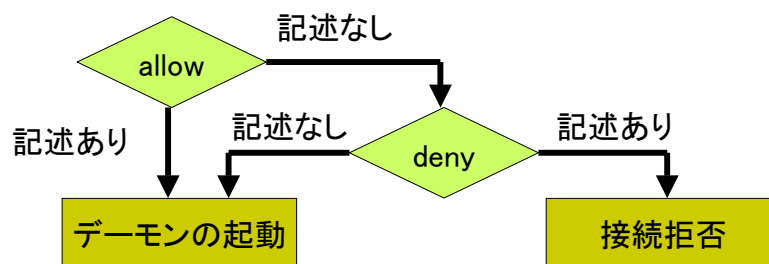
TCP Wrappersの設定

□ 設定ファイルの検索順序

① hosts.allowにあれば許可、なければhosts.denyを見る

② hosts.denyにあれば拒否、なければ許可

※通常は hosts.denyに ALL: ALL と記述しておき、許可するものをhosts.allowに記述



TCP Wrappersの設定

□ 設定例

/etc/hosts.allow

```
in.telnetd: 192.168.0.  
vsftpd: 192.168.0.101 192.168.0.102  
sshd: ALL
```

/etc/hosts.deny

```
ALL: ALL
```

- telnetは192.168.0/24のネットワークからのアクセスを許可
- vsftpdは192.168.0.101と192.168.0.102のみ許可
- sshはすべてのクライアントからアクセス許可
- その他のサービスはすべてアクセス拒否