.Deb

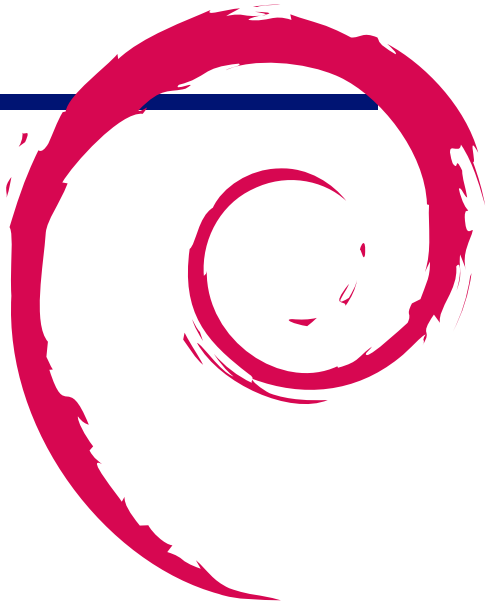
銀河系唯一のDebian専門誌

2024 年 8 月 17 日

WSL2、DebConf24特集

デビアン勉強会

目次



1 最近の Debian 関連のミーティング報告

杉本 典充

1.1 2024 年 7 月度 東京エリア・関西合同 Debian 勉強会

2024 年 6 月 22 日 (土) に東京エリア Debian 勉強会と関西 Debian 勉強会の合同でオンラインによる Debian 勉強会を開催しました。参加者は 11 名でした。

セミナー発表は、林 さんによる「General Resolution:tag2upload は Debian にどんな変化をもたらすのか」、情報交換会「DebConf24 直前特集」を行いました。

勉強会の終了後、参加者同士で Debian や OSS に関する話の情報交換を行いました。

2 事前課題

杉本 典充

今回の事前課題は以下です。

1. WSL / WSL2 のどちらかまたは両方を使ったことがありますか。
2. debconf24 には参加しましたか。
3. debconf24 のトークや現地の話で聞きたいことがあれば、書いてください（任意）。

2.1 dictoss

1. ある
2. 現地で参加した
3. (回答なし)

2.2 NOKUBI Takatsugu (knok)

1. ある
2. ビデオアーカイブを見た
3. (回答なし)

2.3 su_do

1. ある
2. ビデオアーカイブを見た
3. (回答なし)

2.4 23corona

1. ある
2. ビデオアーカイブを見た
3. (回答なし)

2.5 ichienkitte

1. ある
2. 勉強会の報告が初めての参加
3. (回答なし)

2.6 yy_y_ja_jp

1. ない
2. 現地で参加した
3. (回答なし)

2.7 BlueSkyDetector

1. ある
2. 現地で参加した
3. (回答なし)

2.8 kenhys

1. ある
2. ビデオアーカイブを見た
3. (回答なし)

2.9 Hiroyuki Yamamoto (yama1066)

1. ない
2. 勉強会の報告が初めての参加
3. (回答なし)

2.10 testdayo99

1. ある
2. 勉強会の報告が初めての参加
3. (回答なし)

2.11 (=・ω・)ノ (akedon)

1. ある
2. 勉強会の報告が初めての参加
3. (回答なし)

2.12 rsakuma (fubabz)

1. ある
2. 勉強会の報告が初めての参加
3. (回答なし)

2.13 Youhei SASAKI (uwabami)

1. ある
2. ビデオアーカイブを見た
3. (回答なし)

3 WSL2 での sid 生活

佐々木洋平

3.1 はじめに

… あるいは, なぜこんな事になっているのか, 的な.

昨年の夏に Let's Note CF-SR3 を購入しました。ここ十数年はメイン環境としてレッツノートを利用しており, これまでの機材は Debian を常用する機材として特にハマり所も無かったので, いそいそと Debian をインストールしたわけですが...。

1. SecureBoot できません[*1]。
2. Linux 使用時に idle 時の CPU FAN の回転数が下がりません。

特に二つ目が面倒で, ちょっとそのままでは持ち歩けません。周囲の皆様に騒音でご迷惑をかけそうで, おいそれと開くことができません[*2]。

とはいえ, 私の普段の生活は「ターミナル と Emacs(`emacs -nw`)」です。メールの読み書き・コード書きといった日常業務が遂行できないのは非常に困ります。

20 年前ぐらいに VAIO の某製品を購入して X が起動しなかった時には 1 年ぐらい Cygwin 生活をしたのでまたそれかなぁとも思ったのですが, 「WSL2 では systemd が動くようになったらしいよ[*3]」ということなので, それなら WSL2 を試してみるかと, 周辺知識を調べつつ, いそいそとセットアップを開始しました。

... というわけで, 普段使いが unstable な人がどうにか WSL2 で生活環境を整えるまでの顛末についてお話してみます。Debian 固有の話はあまり出てこないんじゃないか (?) と思いますが, 私の様に WSL2 で Debian を使い倒そうと思っている方の御参考になれば幸いです。対応する WSL の version は以下の通りです。

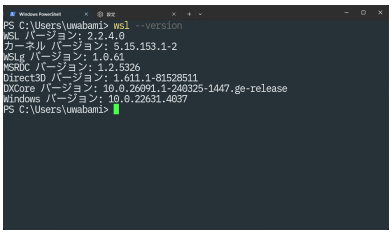


図 1 `wsl --version` の出力結果

[*1] これまで購入してきた CF-S9, CF-SX3, CF-RZ6, CF-RZ8 などは特にハマらず利用できています。BIOS/UEFI の機能が増えたのか, そもそも Debian Installer だと Boot できないんですよね。そのうち時間をとってジタバタしてみたいですが, 一応メイン機材なので格闘する時間がとれず, 現状は諦め気味です。非常に悔しい。

[*2] Panasonic に問い合わせたら, スペックシートとか公開してくれたりしないかしらん。

[*3] `https://devblogs.microsoft.com/commandline/systemd-support-is-now-available-in-wsl/`

3.2 WSL2

「WSL とはなんですか」というお話は本家 Microsoft の記事[*4] なんかで読めるとは思います。ただ, これらの記事は ~~日本語記事が機械翻訳からの翻案になっていて質が良くないのは置いておくとして,~~ 「こうやったら動きますよ」がメインなので, 我々の様な逸般人 (?) が知りたいであろう内部構造について少しまとめておきましょう。

先ず, WSL には WSL1 と WSL2 の 2 つがあります。現在特に何も考えずに利用すると WSL2 が導入されますが, 用途によっては WSL1 を使いたい場合も未だにあるかもしれません。WSL バージョンの比較[*5]にある機能比較表は以下の通りです。

機能	WSL1	WSL2
Windows と Linux の統合	☑	☑
高速の起動時間	☑	☑
従来の仮想マシンと比較して小さなリソース フット プリント	☑	☑
現在のバージョンの VMware および VirtualBox での実行	☑	×
マネージド VM	×	☑
完全な Linux カーネル	×	☑
システム コールの完全な互換性	×	☑
OS ファイル システム間でのパフォーマンス	☑	×
systemd サポート	×	☑
IPv6 サポート	☑	☑

表 1　WSL1 と WSL2 の機能の比較

この表の項目がそれぞれ何を意味しているのかを説明するために, 以下の図を作成してみました。

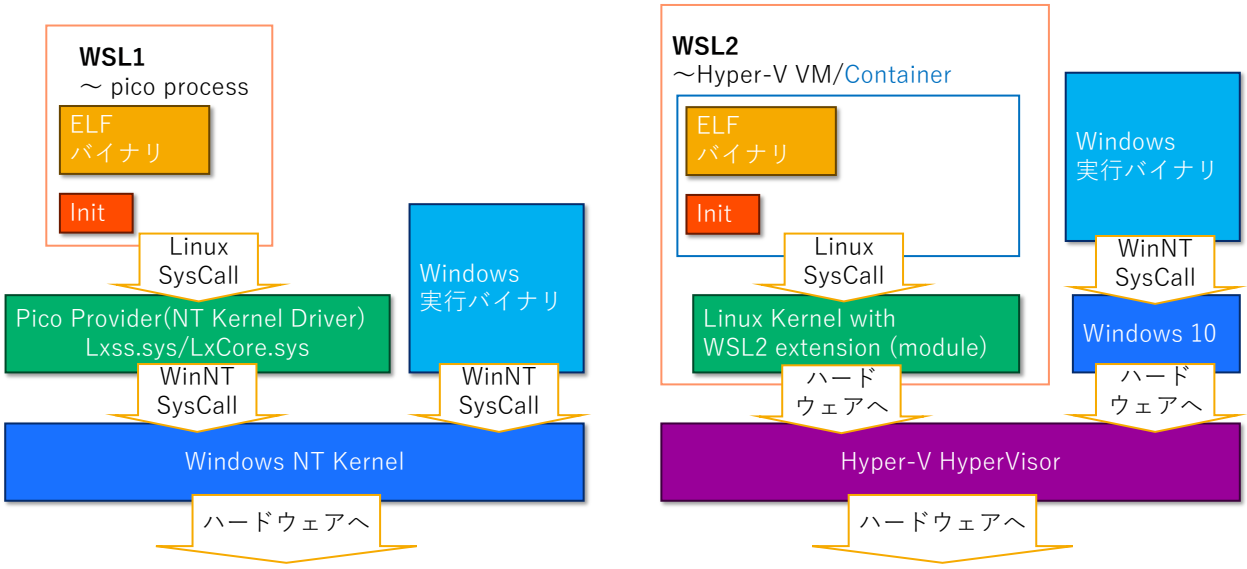


図 2　WSL1 と WSL2 の構造. https://github.com/ionescu007/lxss の資料を元に作成

WSL1 は Windows NT kernel 上で ELF バイナリを動作させるための互換レイヤー (Lxss.sys, LxCore.sys) の上に, Linux ディストリビューションのデータを載せて提供しています。図中の init はディストリビューションの提供している init ではなく, WSL1 が提供している init であり, いわゆる deamon の起動はしません。また, WSL1

[*4] 例えば https://learn.microsoft.com/ja-jp/windows/wsl/ あたり？
[*5] https://learn.microsoft.com/ja-jp/windows/wsl/compare-versions

は VolFS という独自のファイルシステム上にデータが置かれます。VolFS ではファイルの実態は NTFS 上に置かれ, Linux でのパーミッション情報などのメタデータは別途管理されています。Windows 側とのデータのやりとりも DrvFS という独自のファイル形式で管理されています。WSL1 からはドライブレター付きでマウント可能であり, WSL2 との違いとして USB メモリ等もマウントできたりします。なお, DrvFS へのファイルの読み書きは Plan9 由来の 9P server を経由して動作します。9P サーバは WSL1 の提供する init に含まれており, WSL1 起動時に起動しています。

WSL2 は Hyper-V の仮想マシン上で Microsoft が手を入れた Linux Kernel[*6]を動作させることで, ほぼ完全な状態の Linux ディストリビューションが動作しています (していることになっています)。設定次第ですが諸々の daemon も動作させることができます。ネットワークは Hypver-V の仮想スイッチが有効になっており, 初期状態では NAT として動作しています[*7]。また, ファイルシステムは WSL1 とは異なり仮想ディスクを利用します。そのため (オーバーヘッドはあるものの)ext4 やそれ以外のファイルシステムも利用可能です。なお, WSL1 とは異なり DrvFS を利用して Windows 側へアクセスするのではなく Windows 側にも 9P サーバが用意されており, 9P を利用して (ネットワークファイルとして) Windows 側のファイルへアクセスすることになります。

以上, ざっくりメリットデメリットまとめると以下の様になるでしょうか?

バージョン	ベースとなる技術	メリット	デメリット
WSL1	システムコールの変換	メモリ消費が少ない	• 命令変換によるオーバーヘッド • 不完全なシステムコール対応 • ディスクアクセスが遅い
WSL2	Hyper-V 仮想化	• システムコール完全互換 • ディスクアクセス速度の改善	メモリ消費量の増大

表 2　WSL1 と WSL2 の機能比較まとめ

構造が違うので, 現時点では WSL1 と WSL2 は共存できる様です。人によっては用途によって使い分けていたりする様です。

... とはいえ, 自分の用途では

- メールを読むのに mbsync で定期的に手元にメールを取得
- 手元で Dovecot を上げておき, Emacs 上の Wanderlust からメールを読む
- メールの送信は exim4 の smarthost に丸投げ

などといった, daemon を常駐させる様な使い方がほとんどです。そんなわけで WSL2 を導入して設定を開始しました。「WSL2 では systemd が動くようになったらしいよ[*8]」ということなので, きっと大丈夫だろう, という浅はかな考えから始めたのですが...

3.3　ここが変だよ? WSL2

3.3.1　systemd までの道程

導入は Windows の Powershell にて

```
PS C:\Users\uwabami> wsl --install -d Debian
```

[*6] https://github.com/microsoft/WSL2-Linux-Kernel
[*7] 他にもブリッジモードやミラーモードなどがありますが, 上手く使えておりません。
[*8] 再掲: https://devblogs.microsoft.com/commandline/systemd-support-is-now-available-in-wsl/

で OK です。初回起動時にユーザ名とパスワードが聞かれ, ログインできる様になります。

この時点では systemd が動作していませんので, 先ずは `/etc/wsl.conf` を作成し systemd を起動することにします。

```
% vi /etc/wsl.conf
...
% cat /etc/wsl.conf
[boot]
systemd=true
```

その後 WSL を再起動しましょう。

```
$ systemctl status
Failed to connect to bus: No such file or directory
```

アッハイ。

```
$ sudo apt update
$ sudo apt install dbus-user-session
```

ということで, 気をとりなおして WSL を再起動して...

```
$ systemctl status
Failed to dump process list for 'azalea', ignoring: Input/output error
azalea
    State: running
    Units: 280 loaded (incl. loaded aliases)
     Jobs: 0 queued
   Failed: 0 units
    Since: Sat 2024-08-17 12:28:10 JST; 8s ago
  systemd: 252.26-1~deb12u2
  Tainted: cgroupsv1
   CGroup: /
```

ふむ。**Tainted: cgroupsv1** ということで, `/etc/fstab` に以下を追加しておきます。

```
$ vi /etc/fstab
...
$ cat /etc/fstab
cgroup2 /sys/fs/cgroup  cgroup2 rw,nosuid,nodev,noexec,relatime,nsdelegate      0       0
```

その後で再起動して試すと...

```
$ systemctl status
Failed to dump process list for 'azalea', ignoring: Input/output error
azalea
    State: running
    Units: 280 loaded (incl. loaded aliases)
     Jobs: 0 queued
   Failed: 0 units
    Since: Sat 2024-08-17 12:31:44 JST; 22s ago
  systemd: 252.26-1~deb12u2
   CGroup: /
```

... はて。

ということでちょっと調べてみると

- systemd status fails output on Ubuntu #8879 `https://github.com/microsoft/WSL/issues/8879`

にある通りで, 結論から言えば `systemd 255.7` 以降なら大丈夫です[*9]。`/etc/apt/sources.list` を書き換えて, いそいそと sid にアップグレードしましょう。

```
$ sudo vi /etc/apt/sources.list
...
$ cat /etc/apt/sources.list
deb http://deb.debian.org/debian sid main
$ sudo apt update
$ sudo apt-get upgrade
$ sudo apt-get dist-upgrade
```

[*9] WSL のデフォルトが Ubuntu 22.04 のままなのは, こういう事情から, かと思います。

というわけで, 一通り導入が終わったら...

```
  $ cat /etc/debian_version
  trixie/sid
  $ systemctl status
azalea
   State: running
   Units: 308 loaded (incl. loaded aliases)
    Jobs: 0 queued
  Failed: 0 units
   Since: Sat 2024-08-17 12:44:08 JST; 23s ago
 systemd: 256.5-1
 Tainted: unmerged-bin
  CGroup: /
          ├─ init.scope
          │  ├─  1 /sbin/init
          │  ├─  2 /init
          │  ├─  7 plan9 --control-socket 6 --log-level 4 --server-fd 7 --pipe-fd 9 --log-truncate
          │  ├─ 198 /init
          │  ├─ 199 /init
          │  ├─ 200 -bash
          │  ├─ 223 systemctl status
          │  └─ 224 less
          ├─ system.slice
          │  ├─ console-getty.service
          │  │  └─ 185 /sbin/agetty -o "-p -- \\u" --noclear --keep-baud - 115200,38400,9600 vt220
          │  ├─ cron.service
          │  │  └─ 162 /usr/sbin/cron -f
          │  ├─ dbus.service
          │  │  └─ 166 /usr/bin/dbus-daemon --system --address=systemd: --nofork --nopidfile --systemd-activation --syslog-o
          │  ├─ system-getty.slice
          │  │  └─ getty@tty1.service
          │  │     └─ 186 /sbin/agetty -o "-p -- \\u" --noclear - linux
          │  ├─ systemd-journald.service
          │  │  └─ 38 /usr/lib/systemd/systemd-journald
          │  ├─ systemd-logind.service
          │  │  └─ 171 /usr/lib/systemd/systemd-logind
          │  └─ systemd-udevd.service
          │     └─ udev
```

ようやくスタートラインです。

3.3.2 時計合わせ

少し使ってみて気がついたのですが, 起動時はともかく, Host の Windows 側の susupend & resume 後に時刻がズレます。仮想環境あるあるですね。

WSL 側で時刻合わせをしても良いですけれど, Host 側と時刻合わせのタイミングがズレるのはちょっと嫌なので, どうしたモンかな, とすこし悩みました。結論から言えば, WSL2 環境では `/dev/ptp0` が生えていますので, これを使えば良い, ということになります。

- Time sync for Linux VMs in Azure:
  `https://github.com/MicrosoftDocs/azure-docs/blob/main/articles/virtual-machines/linux/time-sync.md`

`systemd-timesyncd` にはまだ `/dev/ptp0` のサポートは無いようです。

- PTP support for timesyncd #22828 `https://github.com/systemd/systemd/issues/22828`

というわけで, 安直には chrony を利用します。上記 Azure のドキュメントにある通り, 設定ファイルに以下を追記して chrony を起動させておきます。

```
% vi /etc/chrony.conf
...
# Step the system clock instead of slewing it if the adjustment is larger than
# one second, but only in the first three clock updates.
# makestep 1 3          # <- コメントアウト
makestep 1.0 -1         # <- 追記
...

## use ptp0             # 以下, 追記
refclock PHC /dev/ptp_hyperv poll 3 dpoll -2 offset 0 stratum 2
```

これで, Windows 側との時刻同期が行なわれる様になる (筈) です。あとは Windows 側で適宜 NTP サーバ等の設定をしておきます。

3.3.3 keyring をどうするか?

ssh-agent, gpg-agent, secret-tools などの, いわゆる「鍵束」をどう使うかに悩みました。Linux の Desktop 環境の場合はログイン認証時に PAM を経由して gnome-keyring の鍵束が unlock されます。これを利用するのが良くある話かとは思いますが, そもそも WSL にはログイン認証がありません。

結局,

- ssh-agent と gpg-agent については `keychain` パッケージを使う
- secret-tools については WSLg があるので gnome-keyring を使う。
  - ログイン時にシェルから呼び出して一度 unlock しておく

といった手段を取る事にしました。

なお **ログインシェルからの呼び出し時の pinentry では何故か GUI のキー配列が US 配列になります**。常に US 配列となるわけではなく, 2 回目・3 回目の pinentry の呼び出し時には (言語か何かを見たのか) 日本語配列となります。... なんでですかねぇ。

3.3.4 快適なターミナルを求めて

快適にターミナルで生活するための条件は人によって異なるとは思いますが, 私の場合は

- East Asian Ambiguous Width が制御できること。
- Emoji がちゃんと表示できること。特に Zero Width Joiner をちゃんと処理できること。

が上げられます。前者はターミナルと Emacs の文字幅を揃えるために, 後者は Weechat などで崩れずに表示するために, でしょうか[*10]。現時点では

- 表示が一番良いのは WSLtty (mintty の WSL 対応版)
- 表示がマシで, 処理速度が軽快なのは WezTerm

ということで, 基本的に WezTerm を利用しています。とはいえ WezTerm の公式ドキュメントにある通り[*11], WezTerm をそのまま起動して wsl を叩くと, 折角の文字処理が全て ConPTY に持っていかれて悲しい目に合いますので, WSL2 を起動したら, Wezterm 内蔵の SSH で接続して利用しています。sshd の設定で `AcceptEnv` を

```
$ cat /etc/sshd_config
...
# Allow client to pass locale environment variables
# AcceptEnv LANG LC_*
AcceptEnv LANG LC_* TERM COLORTERM TERM_PROGRAM TERM_PROGRAM_VERSION WSL* WEZTERM*
...
```

と幾つか追加しておくと良いでしょう。

3.4 おわりに

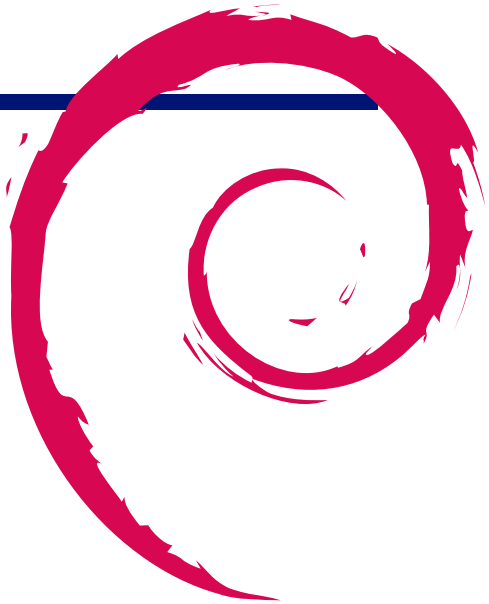
というわけで, 生活環境としての WSL2 の設定でハマった所について簡単にまとめてみました。動作させ始めてからは, まあ普通の環境かな, 学生に実習などで利用してもらう上では非常に便利かな, とか思いつつも, ネットワーク周りがまだまだちゃんと整理できておらず, ちょっとモヤモヤしていたりします (特に VPN 利用時のネットワーク周りの設定が良くわかっていません)。

WSL2 はこれからまだまだ更新されていくとは思いますので, 現時点での情報として御笑覧頂ければ, 幸いです[*12]。

[*10] ついでに, 動作が軽快であるとか, 余計な処理をしないとか, 余計な装飾をしない, とか。イロイロ求め出すとキリが無いんですが, まあそれはそれとして。

[*11] Windows and ConPTY: `https://wezfurlong.org/wezterm/what-is-a-terminal.html?h=conpty#windows-and-conpty`

[*12] なお, 別に WSL2 が好きな訳じゃないくて, ちゃんと動作させられるなら素の Debian 環境に移行したいな, とは思ってます。次に買うならなんだろう。やっぱり ThinkPad かな。X1 Nano Gen3 って Linux 動くんだろうか?



4 メモ

Debian 勉強会資料

2024 年 8 月 17 日　　初版第 1 刷発行

東京エリア Debian 勉強会 （編集・印刷・発行）