

VDI 環境構築作業時に遭遇した Windows 的「へえ～」

情報社会基盤研究センター担当

中野 裕晶

情報社会基盤研究センターでは、2014年3月から Windows 7 及び 8 による VDI 環境のサービスを開始した。その準備作業を行ってきた中で、それまで知らなかった Windows の動作等に関する情報を幾つか載せておく。

1. Windows7 でのネットワーク速度改善

Windows7 において、ネットワーク速度がやたら遅いという現象に遭遇した。調べてみると、巷では Windows7 のネットワークが遅いと感じられる事例がそれなりにあるらしく、対策法として、SNP の無効化や自動チューニングレベルの変更といったものが紹介されていたので、実際に試してみた。

まず、現在の設定内容を

```
netsh int tcp show global
```

コマンドで確認し、その後、

```
netsh int tcp global rss=disbale
```

```
netsh int tcp global chimney=disbale
```

```
netsh int tcp global netdma=disbale
```

```
netsh int tcp global autotuninglevel=restricted
```

コマンドで設定変更を行った。設定が変更されたどうかを再度、

```
netsh int tcp show global
```

で確認。autotuninglevel の値には、一般に highlyrestricted か restricted が効果的とされているらしく、今回試した環境では、restricted で改善が見られた。

ちなみに、ネットワーク速度が遅かった現象については、最終的に上流のネットワーク機器変更で完全に解消されてしまった事を付け加えておく。

2. フォルダ名の「.」使用

Windows では、「.」を使ったフォルダを使用することができる。しかし、この「.」に全く対応していない、或は一部機能が使えないアプリケーションが存在した。

情報センターがサービスしている Windows 環境において、AppData、Desktop、My Documents 等についてはフォルダリダイレクト機能を利用しており、ユーザのファイルはファイルサーバへ保存されるようになっている。このリダイレクト先のパスには「.windows」というフォルダが含まれており、デスクトップやマイドキュメントにあるファイルを開けない、保存できないという状況であった。取り敢えず、暫定的な対処方法によって問題は回避されているが、フォルダ名に「.」を使う事は避けた方が良いのかもしれない。

3. ショートカットとリンク (シンボリックリンク、ハードリンク、ジャンクション)

UNIXの世界では、シンボリックリンクやハードリンクという機能がある。これら機能のWindows版がショートカットなのだろうと思い込んでいたのだが、実はWindowsにもシンボリックリンクやハードリンクといった機能が存在していることを知った。

ショートカットを使用した場合、その拡張子に「lnk」がついてしまう。例えば、「.¥LINK¥Folder」というパスでアクセスできるようにしたい場合、ショートカットを用いて実現しようとする

「.¥LINK.lnk¥Folder」となってしまう目的を果たせない。シンボリックリンクを使用すれば、望み通り「.¥LINK¥Folder」としてアクセスできるようになる。

リンクを作成するコマンドは、

```
mklink [オプション(/D /H /J)] リンク元 リンク先
```

であり、UNIXコマンド(ln [-s] リンク先 リンク元)と比べて、リンク元、リンク先の順番が逆となる。オプションを指定しない場合は、シンボリックリンクが作成されるということになっているが、ちゃんと機能しない場合がある(Windowsのバージョンによる?)ので、シンボリックリンク作成する場合は、/Dオプションを付けて、確実にシンボリックリンクと指定することをお勧めする。

もう一点、シンボリックリンクを使用する時に気にする事がある。それは、リンク元とリンク先がそれぞれローカルディスクにあるかリモートディスクにあるかで、シンボリックリンクが使えるかどうかの境界がある。この境界は設定変更可能であり、現在の設定がどうなっているかを確認するには、

```
fsutil behavior query symmlinkevaluation
```

を実行すればよい。デフォルトの設定は、リンク元がローカルディスクにある場合のみリンクが有効となっている。設定変更は、

```
fsutil behavior set symmlinkevaluation 値
```

というコマンドでもグループポリシーでも可能である。

4. Virtual Store

Windows7にTeraPadを新規インストールして使用してみた。最初にファイルを保存しようとした際、保存先として指定されるのは、C:¥Program Files¥TeraPadフォルダである。「えっ? こんな所に一般ユーザでの書き込み権限があるの?」と思いつつ、素直に保存を試みると保存に成功。ならばと思い、エクスプローラでC:¥Program Files¥TeraPadにアクセスしてみると、つい今しがた保存したファイルが見当たらない。さらに、このフォルダに新規にフォルダを作成してみようとする、見事に失敗。再度TeraPadを起動し、TeraPadから先程保存したファイルを開こうとすると開くことができる。

はて? 保存したファイルは何処へ?

答えは、C:¥Users¥%username%¥AppData¥Local¥VirtualStore¥Program Files¥TeraPadであった。

これは、Windows Vistaから導入されたVirtual Storeという機能によるものだそうだ。

この場所に保存されたファイルは各マシンのローカルディスクに保存されてしまう為、ログアウト後に

使用していた仮想マシンが削除される環境下では、ファイル消失といった問題が起こる可能性がある。Virtual Store の機能停止は、グループポリシーやレジストリの設定で行えるのだが、この機能を無効にすることで逆に何か問題が起きないか確認が必要である。

ちなみに、Virtual Store 機能は、C:\Program Files の他に、C:\ProgramData と C:\Windows も対象のフォルダとなっているらしい。

5. ACL の挙動

自分が UNIX のパーミッションに慣れてしまっている為か、Windows の ACL の挙動に違和感を覚えた。

/AAA/BBB というディレクトリがあったとする (Windows 的表記だと、C:\AAA\BBB)。

UNIX の場合、ディレクトリ AAA に対して 700 というパーミッションの設定を行うと、オーナー以外は AAA 配下のディレクトリやファイルには一切アクセスできない (拡張属性使用時の挙動は未確認)。

しかし Windows では少々挙動が違っていた。フォルダ AAA に対してユーザ A のみにフルコントロール権限を与え、サブフォルダ BBB に対してはユーザ B へのフルコントロール権限を追加したとする。ユーザ B がエクスプローラを使用して AAA を開こうとすると、当然開く事はできない為、BBB の下まで辿り着くことはできない。しかし、ユーザ B がアドレスバーに C:\AAA\BBB と直接パスを記述すると、フォルダ BBB の下のファイル、フォルダにアクセスできてしまう (親フォルダ AAA の ACL は無視されてしまう) のであった。