Vista 対 ほげ

文責:Nidhoggr

第一章：Vista？ なにそれしらねーよ

どうも、ほげです。もはや大学生活も五年目に入りましたが未だ学部生です。あっれー？[1]

去年は boost について書いてお茶を濁した気がしますが[2]、今回はバイトで Vista と闘ったときの闘病生活、もとい戦闘に関する記録を少し書きなぐってみたいと思います。

そもそもほげは Vista を持っていない&買う気も使う気も無かったのでしばらくは無縁だと思っていたのですが、バイト先がまさにその Vista 対策に喘いでいたのでしょうがなくおいらも戦闘に参加することになりました。

「うへへへｗｗｗまあなんとかなりますおｗｗ」

これが罠の始まりであった。

第二章：Session0 という罠

第一話「対話型・非対話型」

まず本題に入る前に、読者諸君は Windows アプリケーションにおいて**「非対話型」**と**「対話型」**のアプリケーションがあるということを知っているでしょうか？

これはアプリケーションがユーザーの操作を受け付けるかどうかという違いです。

「対話型」とはマウスやらキーボードなどで操作をした結果が返ってくるアプリケーションを指します。Word だとか Sleipnir だとかが該当します。アプリケーションと言ってほとんどの人が思い浮かべるのがこのタイプです。

それに対し、**「非対話型」**のアプリケーションは基本的にユーザーからの入力を受け付けず、システムから受けたレスポンスに対して応答します。Windows の裏側でセコセコと動いてらっしゃる**「Windows サービス」**と呼ばれるアプリケーション群がこれにあたります。

第二話「CreateProcess」

Windows には**CreateProcess** という API があります。これはその名の通り新たなプロセスを立ち上げるもので、窓アプリ開発者ならば一度は触ったことがあるかと思われます。

プログラム上から違うプログラムが呼べるので非常に便利なのですが、上記の非対話型・対話型間でこれをやろうとするとかなり面倒です。

そもそも Windows の仕様として非対話型と対話型のアプリケーションを相互にプロセス生成ということが出来ないようになっています。非対話型で作ったプロセスはすべて非対話型に、対話型で作ったプロセスはすべて対話型になります。Windows サービスから Sleipnir 等の対話型プロセスを立ち上げるとプロセスは立ち上がりますがウィンドウが表示されず。操作することも不可能です。

では非対話型から対話型プロセスは作れないのかというとそうではなく、実行中のプロセスを非対話型から対話型へ、もしくはその逆に変更する API が存在するので、一度変換してから CreateProcess を呼べば問題ありません。但し安全のために非対話型・対話型に直す必要があります。

第三話「罠」

「なーんだ……簡単じゃねえか」

と、思うでしょう。確かにこのようにすれば XP までならば問題はありません。XP までならば。

---

[1]五年制の大学なんです、きっと

[2]boost::serialization について書いたな

そう。問題は Vista なのです。ここで Vista と XP の違いが議論に上がってきます。

そもそも XP の構造として、**Session** という概念があります。これはユーザー管理上の概念で、**「ログオン・セッション」**というと分かり良いかと思われますが、すなわち Windows にログオンしてからログオフするまでにユーザーにあてられる単位です。

たとえば立ち上げたばかりの XP にユーザー「hoge」がログオンすると、ログオンユーザーである hoge には「Session0」というセッションが与えられます。以降、hoge がログオフするまで Session0 は hoge のセッションとして稼働します。ここに新たなユーザー「piyo」が入ってきたとしたら piyo には「Session1」が割り当てられ、以降それが続きます。この Session0 と Session1 との間は隔絶されているので、Session0 から Session1 上で動くプロセスを立ち上げることは原則できません。

これが Session の概念です。XP ではこの Session0 や Session1 にそれぞれのユーザーの対話型・非対話型アプリケーションが動いています。

しかし、Vista では Session0 を特別なセッションとして、非対話型アプリケーションはすべてこの Session0 にまとめ、Session0 には対話型アプリケーションは置けないようになっています。これが「Session0 分離」であり、これの影響によって Vista では Windows サービスプログラムからは対話型アプリケーションが起動できなくなっています。これにはおいらも非常に難儀しました[3]。

第三章：Let's Vista FHuck!!

この Session0 分離でもはや望みは尽きたのかと思われましたが、そこは回避策が一応用意されていて、しかもそれは M$さんが推奨している方法なのですが、非常に場当たり的というかやっつけというか、なんだか予め用意した道には見えないのですが……[4]

まあその回避策として用意されている(？)のが WindowsTerminalServer(WTS) です。これは本来だと一台のサーバに複数端末からリモートログインするための機構であり本来ローカルで利用するものではないのですが、誰かがログオンした瞬間にすでにマルチログインのような感じになってしまう Vista においては大変有用です。

で、これを使ってどうするかというと、WTS の API を使ってログオン中のセッションの ID を取得し、そのセッション ID のユーザーを偽装してプロセスをリモート起動するという、ハッキングまがいのことをします。どう頑張っても公式推奨の方式とは思えん。

これらの API 名は失念しましたが、「WTS セッション ID」あたりの検索ワードでググって頂ければ幸せになれるかもしれません。

第四章：闘い・その後に

というわけで Windows Vista で Windows サービスプログラムを立ち上げ、そのサービスからアプリケーションを立ち上げるという機構を作ることができました。冷静に考えると酷いアプリケーションだな・・・[5]。

Vista 対応は本当に非常に面倒かつ大変で血反吐を吐く作業です。どこの企業もウンザリしていることでしょう。読者諸君でこれから Vista 対応をしようとしている人、もしくは既にやっている人、まさに今対応中の人は地獄の道ですが頑張ってください。

---

[3]先の見えないデスマが一番嫌いだ
[4]むしろ考えていなかったとしか思えない
[5]どこのウィルスですか？