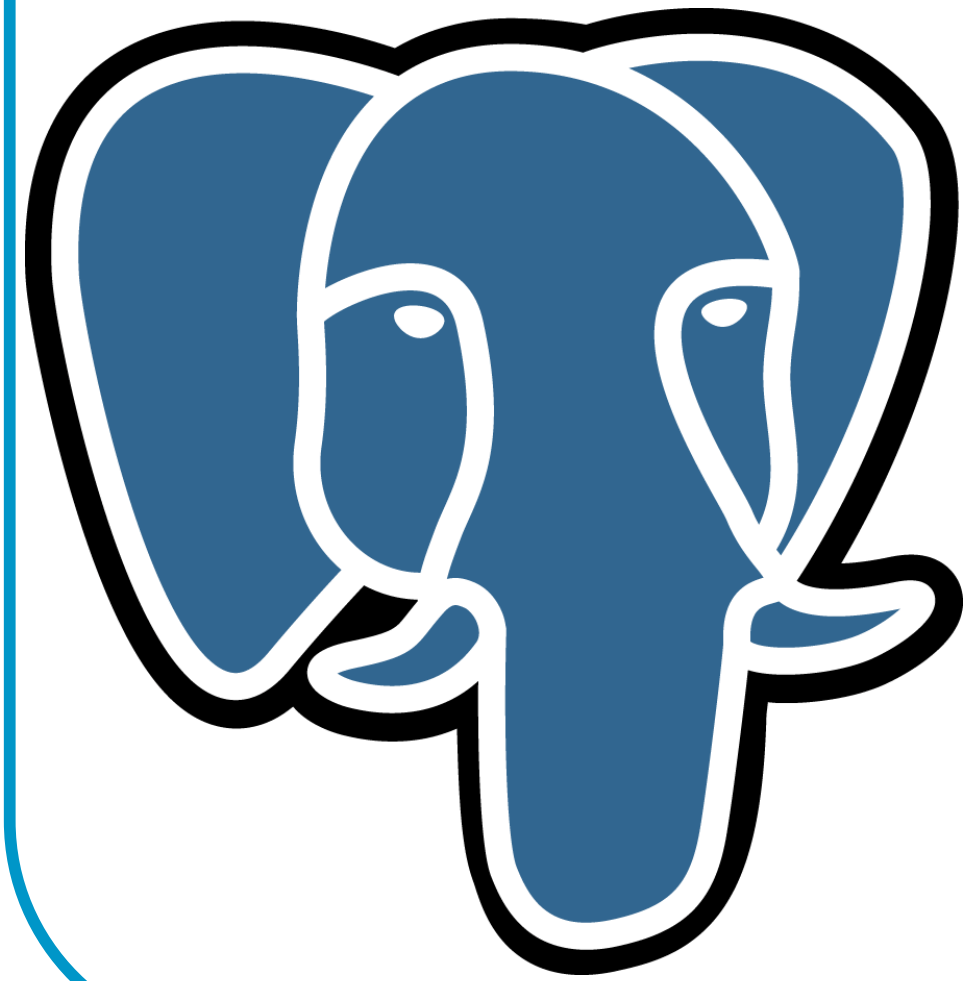
PostgreSQL on Windows



Magnus Hagander
magnus@hagander.net

PostgreSQL Global
Development Group

Agenda

- Why PostgreSQL on Windows
- PostgreSQL for the Windows user
- Windows for the PostgreSQL user

Why PostgreSQL on Windows

- Isn’t Linux better?
  - Often, but not always
- Several scenarios
  - Developer laptops
  - Desktop database
  - Migration scenarios
  - Corporate policies

Agenda

- Why PostgreSQL on Windows
- PostgreSQL for the Windows user
- Windows for the PostgreSQL user

PostgreSQL for the Windows user

- Unix inheritance
  - Multi-process, not multi-thread
  - Shared memory
  - Requires ”modern” windows (2000+, no FAT)
- Commandline!
  - psql, pg_dump, pg_dumpall, etc
  - pgAdmin3 to the rescue!

PostgreSQL for the Windows user

- Environment variables
- Configuration files
  - No registry
  - Edit with any text editor (e.g. notepad)
  - Edit with pgAdmin3
    - Still just a textfile
  - Signal server to reload (pause service)
- Security
  - Will *not* run with administrative privileges

Agenda

- Why PostgreSQL on Windows
- PostgreSQL for the Windows user
- Windows for the PostgreSQL user

Windows for the PostgreSQL user

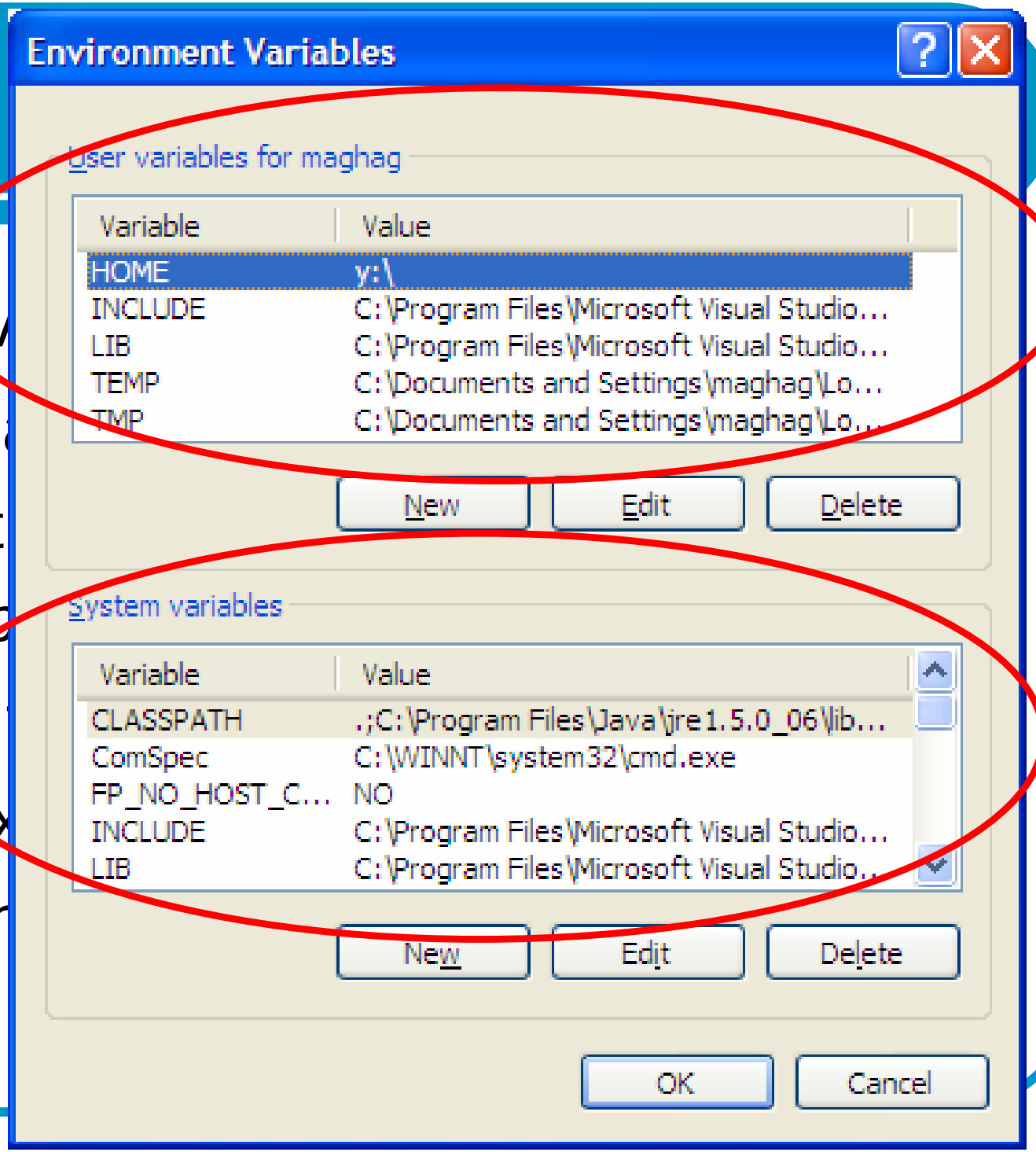
- It’s a brave new world
- None of the normal tools
  - No ps
  - No kill
  - No top
  - No cron
- Builtin Windows tools are generally bad

Architectural differences

- Runs as a service
  - Windows version of daemon
  - Started by ”Service Control Manager”
  - Has it’s own login
- Logs to a file or eventlog
  - Startup errors *always* to eventlog
- CreateProcess() instead of fork()
  - Extra important to use connection pooling

Installing

- Use the M
- External l
  - Perl, pyt
  - Path is *p*
- Or install
  - Complex
- Multiple ir



Configuration parameters

- shared memory
  - Workload dependant
  - Smaller is better?
- fsync methods
  - open_datasync
  - fsync_writethrough
- log_destination
- lc_xyz

Managing PostgreSQL

- Get the right tools!

  http://www.microsoft.com/technet/sysinternals

  - Process Explorer (*the* tool for Win32)
  - pstools
  - Process monitor (debugging)
  - Junction (tablespaces, xlog location)

Where did it go – ps title?

- Unix:

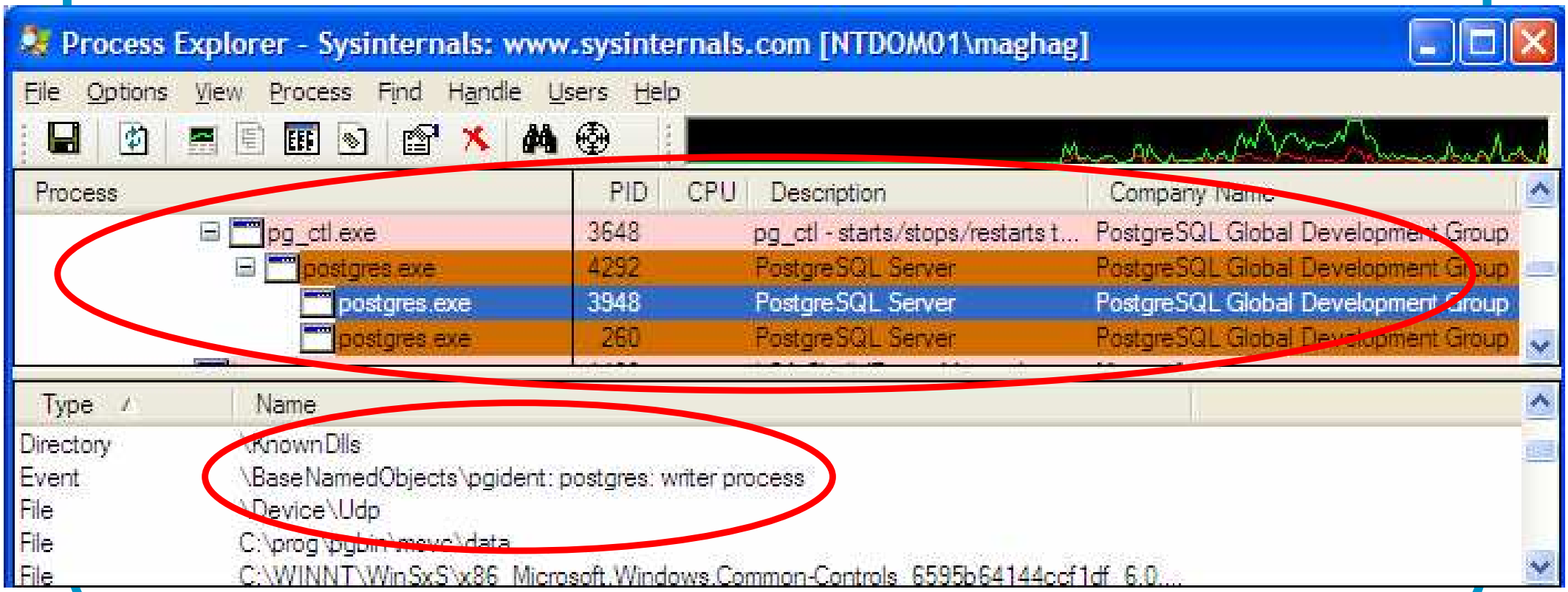
```
root@svr1:~# ps axuwf |grep postgres
pgsql     1397  0.0  0.1  5360 1744 ?        S    2006   0:23  \_ postgres: logger process

pgsql     1399  0.0  0.3 15708 4160 ?        S    2006   2:29  \_ postgres: writer process

pgsql     1400  0.0  0.1  6360 1808 ?        S    2006   0:03  \_ postgres: stats buffer process
```

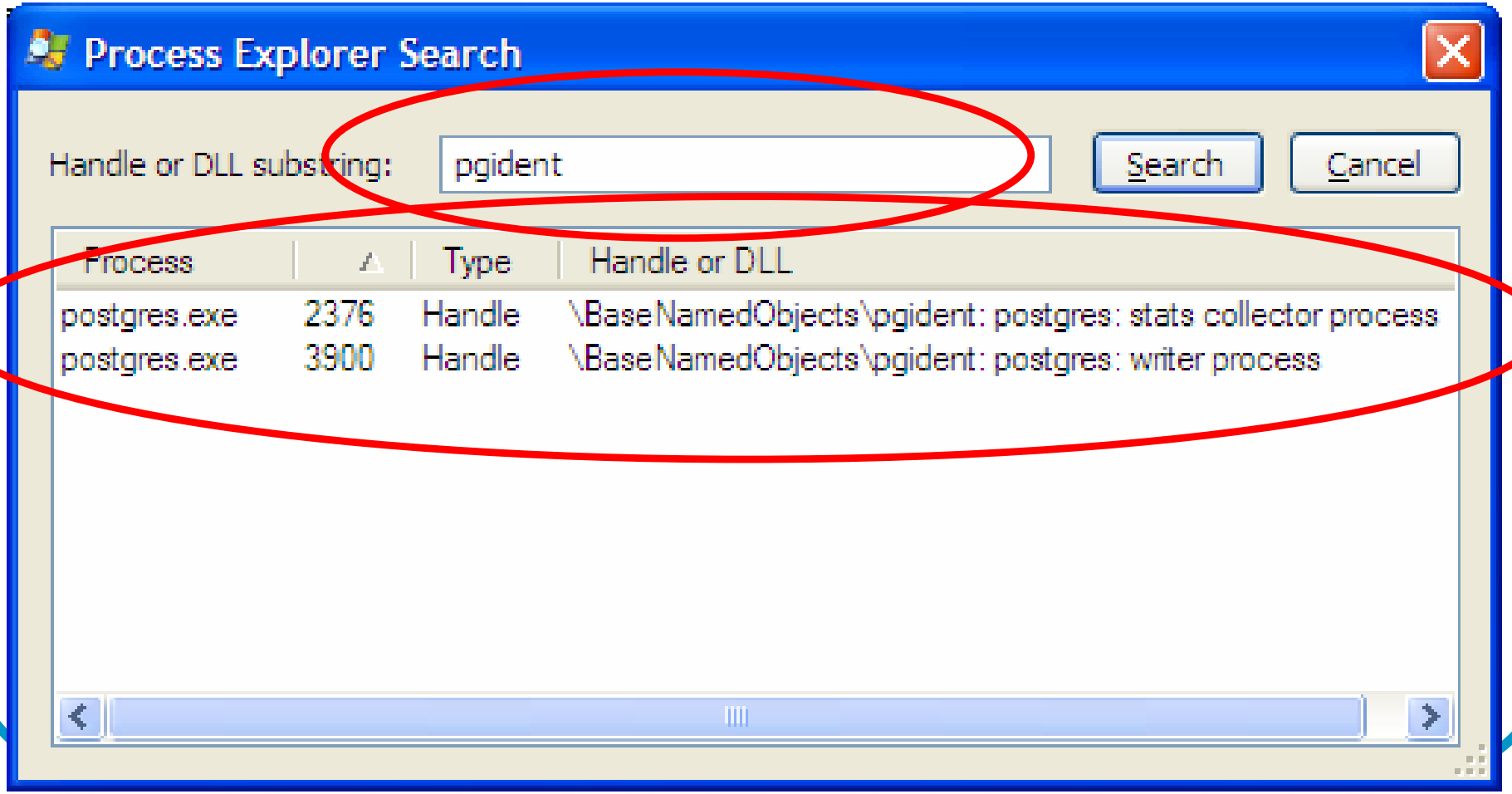
Where did it go – ps title?

- Win32:



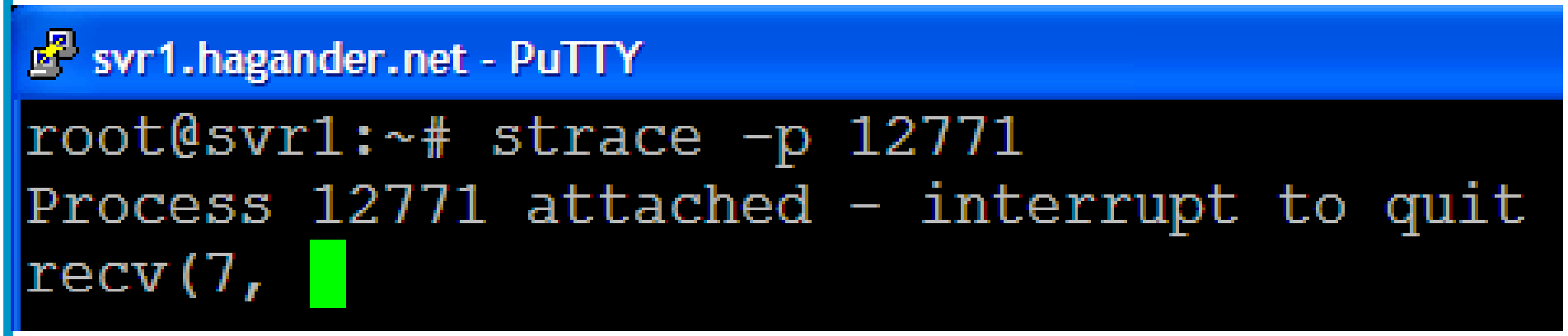
Where did it go – ps title?

- Win32:



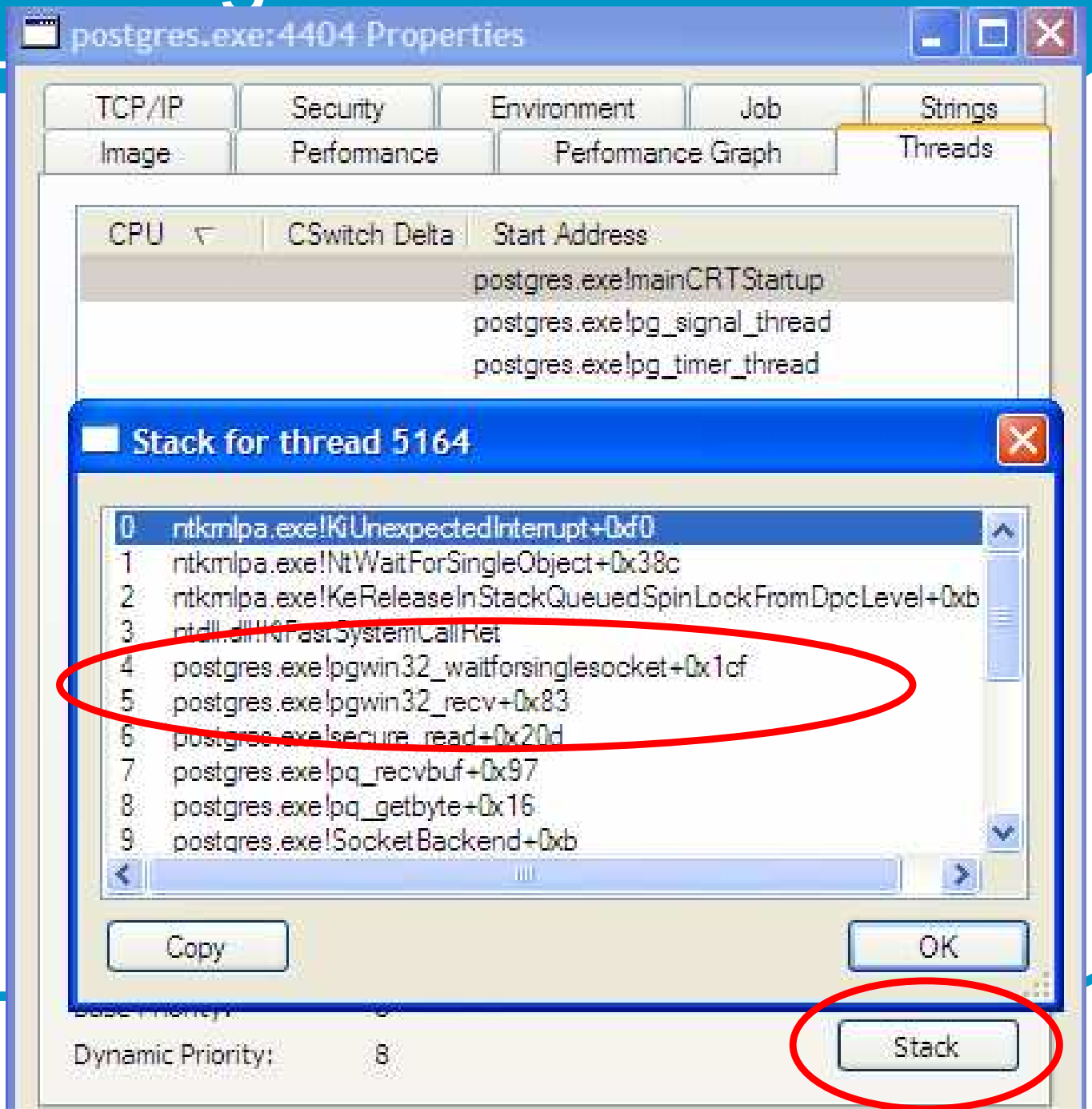
Where did it go – strace

- Unix:



Where did it go – strace

- Win32:



Monitoring PostgreSQL

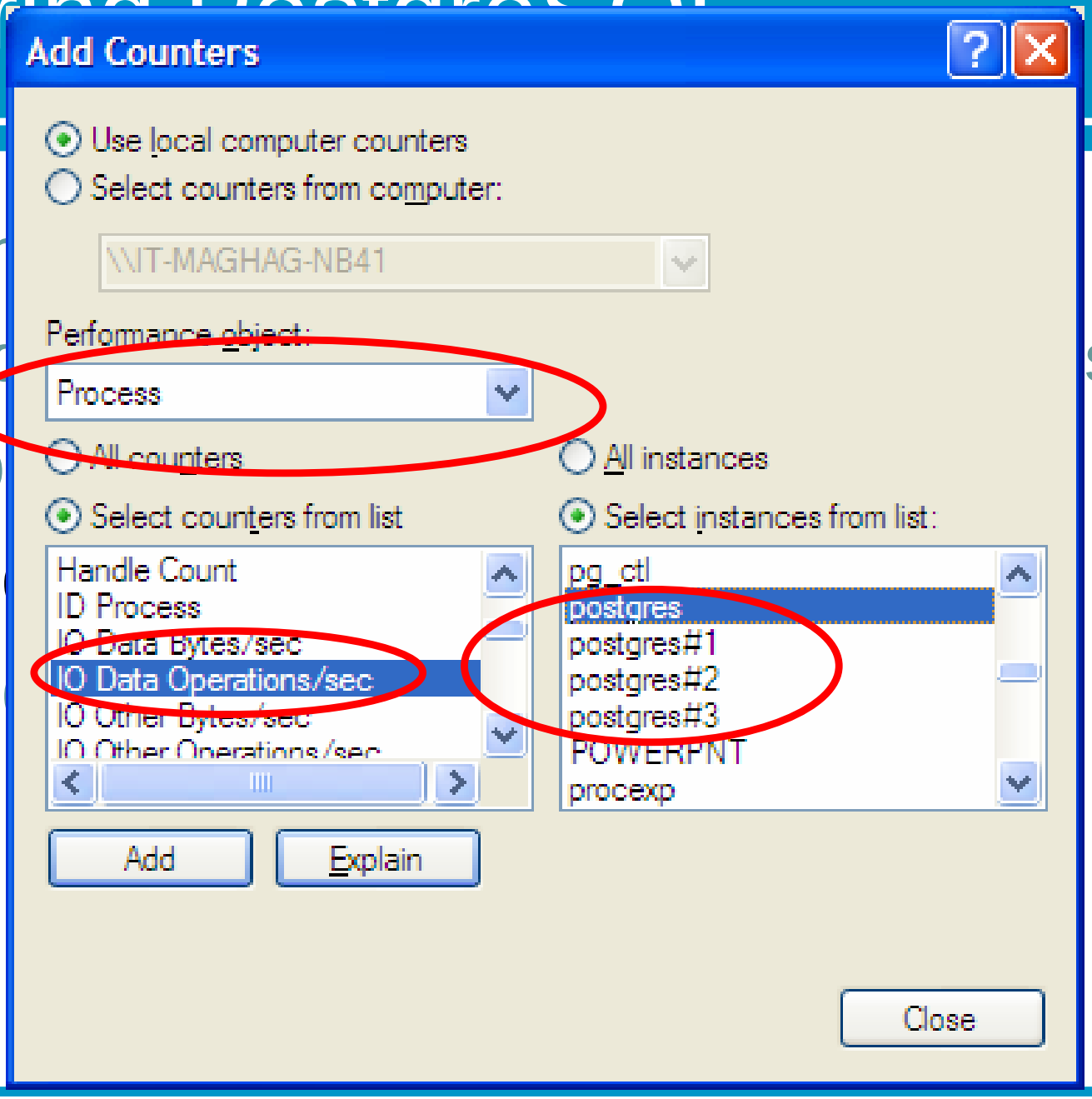
- Some things really didn’t change
- pg_stat_xyz
  - *Has not changed at all*
- PostgreSQL logs
  - pg_data*.log
  - Don’t forget eventlog!

Monitoring PostgreSQL

- Performance Monitor
  - Standard Windows monitoring still applies!
- Monitoring the whole server
- Looking at individual backends
- Looking at the whole cluster

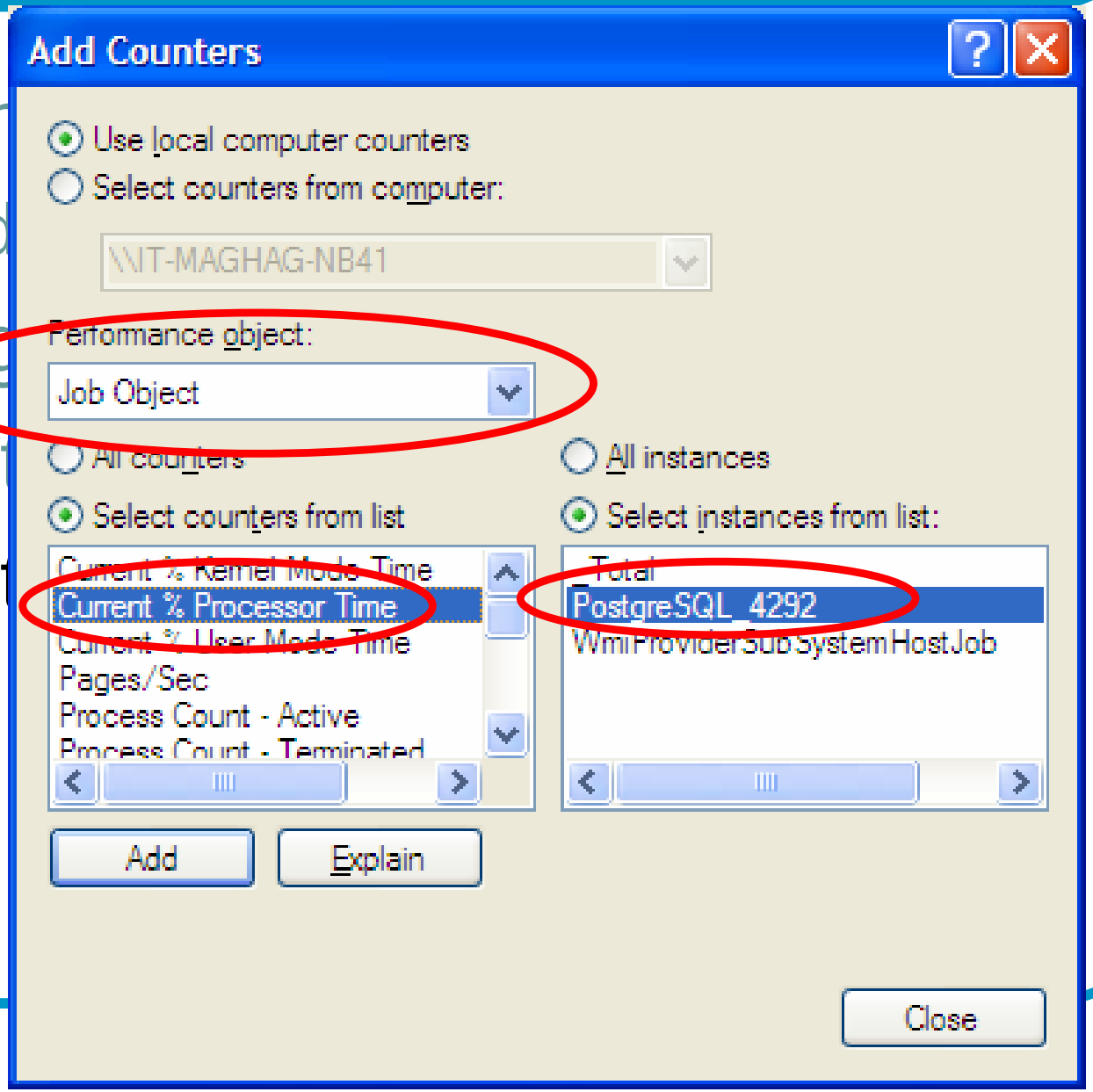
Monitor

- Perform
  - Stand
- Monito
- Lookin
- Lookin



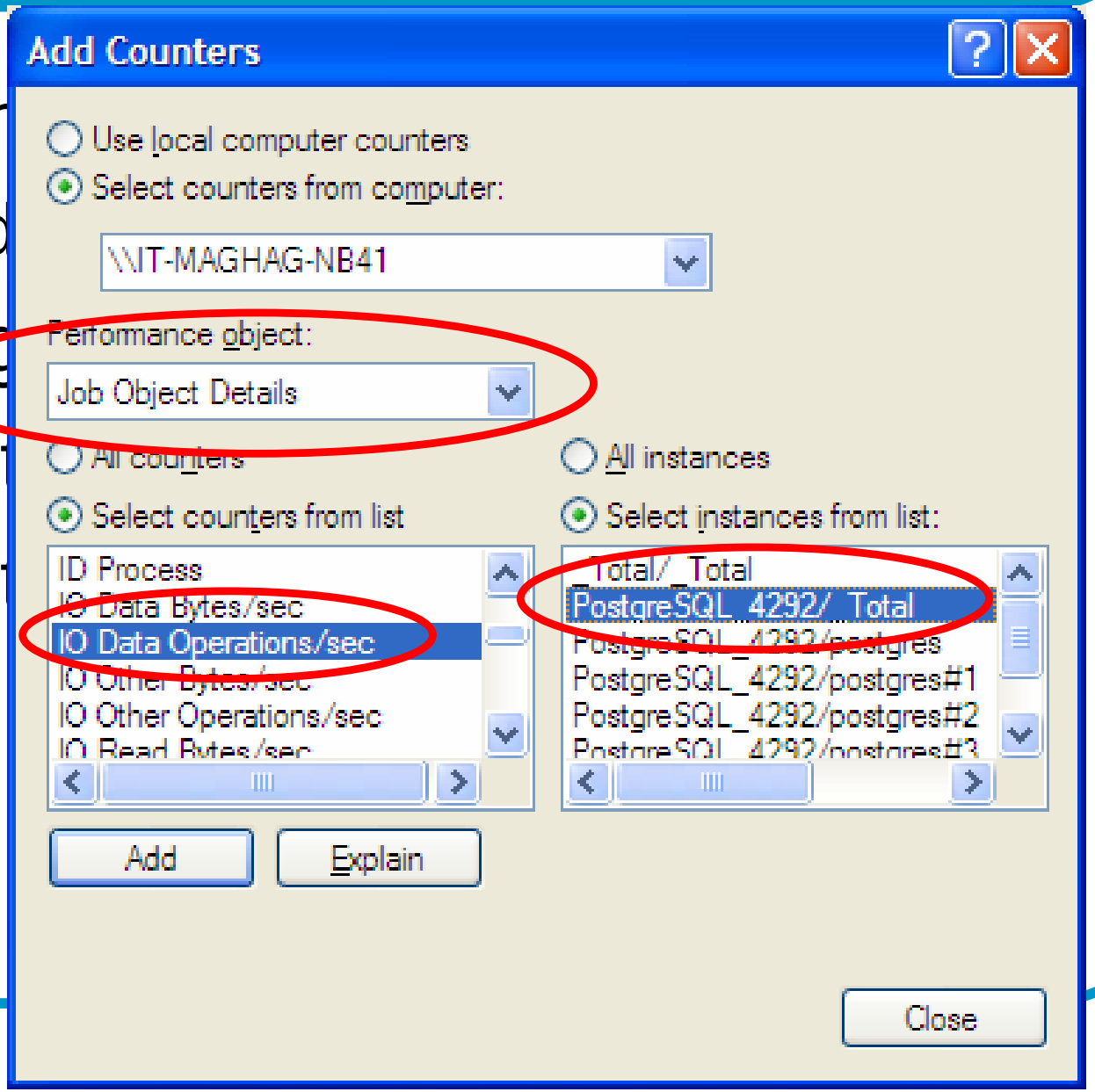
Monitoring PostgreSQL

- Performar
  - Standard
- Monitoring
- Looking at
- Looking at



Monitoring PostgreSQL

- Performar
  - Standard
- Monitoring
- Looking at
- Looking at



Interesting counters to watch

- Number of processes
- New process creations
- Pages/second
- Memory usage (private bytes, working set)
- I/O operations / second
- I/O bytes / second
- Physical disk queue length

Monitoring with Process Explorer

- Add extra columns
  - Private bytes
    - Local backend memory
  - WS Shareable / WS Shared
    - Shared buffer + shared code
  - Virtual Size
    - Just address space
  - Context switch delta
    - Expensive on Windows!

Thank you!

Questions?