Desktop Linux Kernel project

Desktop Linux Kernel project

Krzysztof Lichota
<lichota@mimuw.edu.pl>

What Desktop Linux Kernel Project is

- Project to improve desktop usage of Linux at the level of operating system or in related areas, i.e:
  - improving kernel
  - improving system low-level libraries
  - improving interaction between kernel and desktop environment/system libraries
  - improving general system layout

What Desktop Linux Kernel project is not

- Another desktop project (like KDE/Gnome)
- Another Hardware Abstraction Layer
- Kernel/desktop environment bugfixing project (unless you consider improving performance/functionality/etc. a bug per se)
- Driver writing/improvements project

Rationale

- Desktop Linux issues are frequently ignored in favour of server issues (e.g. swap-back patches)
- Projects/patches to improve desktop experience are scattered across many people/companies.
- There is currently no group which tries to improve desktop experience in a systematic way.

Goals

- Investigate various desktop issues
- Evaluate existing solutions using scientific methods or create new solutions
- Group all desktop patches/fixes/improvements in one ready-to-use package for interested parties (users, distributions)
- Provide recommendations/guidelines to distributions how to improve desktop experience.

Areas of research

- Performance
- Desktop functionality improvements
- Security
- Power management

Performance

- Investigation of main bottlenecks in desktop usage and fixing them:
  - Disk access characteristics
  - Swap usage
  - Scheduling behaviour
  - Memory usage patterns
  - Special desktop scenarios

Disk access characteristics

- Disk access latency
  - Prefetching
  - Using low-latency devices for caching (ReadyBoost)
- Disk access locality
  - Grouping frequently accessed files
  - Reordering partitions for maximum performance (e.g. swap between / and /home, putting swap in file on /, ...)

Swap usage

- Investigate swap usage patterns
- Suggest and implement improvements
- Provide recommendations for swap usage (swap size, swap location, tunables values)
- Evaluate existing solutions (e.g. swap-back patches by Con Kolivas, swap compression patches)

Scheduling

- Create sensible metrics for evaluating schedulers or use other scientifically-proven methods (user tests using double-blind experiments)
- Evaluate existing schedulers using these methods
- Improve existing schedulers
- Provide sensible values for scheduler tunables

Memory usage patterns

- Evaluate usage of RAM by applications
- Suggest possible improvements (e.g. pinning some memory, providing applications with minimum allocated RAM guarantees, working-set based approaches, evaluate swap-out of whole applications, etc.)

Special desktop scenarios

- Improve speed of:
  - Hibernation/unhibernation
  - Suspend/unsuspend
  - Login/boot
  - Logout/shutdown
  - User switching

Desktop functionality improvements

- Partitionless desktop (partitions as files)
- Easy mounting of remote protocols (universal KIO-FUSE)
- USB pendrive removal (Maciej Kowalczyk work)

Security

- Evaluation of performance/security of various security measures (SELinux, AppArmor, jails, ...) and providing recommendations to distros
- Encrypted files/disks/desktops
- Providing integrated package of security measures suitable for desktop users (address-space randomization, ...)
- DNS securing (Rafał Krypa work), firewalling

Power management

- Evaluation of main power bottlenecks
- Applying and enabling useful features (dynticks)
- Providing sensible tunables (hard-drive turnoff time, CPU scaling behaviour, etc.)

Others

- Virtualization for security
- Hibernation – preventing reusing swap, booting with other system version
- IO scheduling
- Resource priority for foreground tasks
- OOM killer prevention
- Thrashing prevention

Others (2)

- Task manager (killing) protection
- Dynamic swap
- Force unmount (error to application)