Make livepatch callbacks, shadow variables and states work together

Petr Mladek <pmladek@suse.com>
Kernel Developer at SUSE
Livepatch features

- Using ftrace to redirect functions (basic)
- 4 callbacks: pre-patch, post-patch, pre-unpatch, post-unpatch (optional)
- Shadow variables to extend structures, global variables
- States for managing the lifetime and compatibility of callbacks and shadow variables
Livepatch structure
Livepatch structure

Patch
Livepatch structure

- Patch
- Object
Livepatch structure

- Patch
- Object
- Func
Livepatch structure

- Patch
- Object
- Func
- Callback
Livepatch structure

- Patch
- Object
- Func
- Callback
- Shadow Variable
Livepatch structure

- **Patch**
- **Object**
- **Func**
- **Callback**
- **Shadow Variable**
- **State**
Livepatch structure

- Patch
- Object
- Func
- Callback
- Shadow Variable
- State
Livepatch structure

- Patch
- Object
- Func
- Callback
- Shadow Variable
- State
Livepatch structure

- Patch
- Object
- Func
- Callback
- Shadow Variable
- State
Livepatch structure

- Patch
- Object
- Func
- Callback
- Shadow Variable
- State
More Changes

- Calling callbacks when using atomic update:
  - Current: only from new livepatch
  - Newly: when the state is introduced or removed (safe downgrade)
- Garbage collection of shadow variables (simple)
- Replace “state→version” with “state→block_disable”
Summary

- Allows to manage callbacks and shadow variables reasonable way
- Per-object callbacks are no longer supported (rare, custom notifiers)
- Livepatch API changes (patches are version specific anyway)

POC: https://lore.kernel.org/lkml/20231110170428.6664-1-pmladek@suse.com/
Thank you