

# Shared Memory for the LHC

This article is meant to document the work in progress regarding the shared memory usage in the future LHC. This project is part of the multi-core R&D effort at CERN.

## Child Documents

The following documents are associated to the shared memory effort:

SharedMemoryLhc

- SharedMemoryAllocatorTests
- SharedMemoryTests
- SharedMemoryTools

## Lessons Learned

The study of shared memory has allowed us to learn the following things:

- Shared memory can not be used transparently by the client code. There are two approaches: A) create a Factory, B) create a base class that allocates the object in shared memory.
- "template typedef" tricks.
- A Factory implementing RAII simplifies client code.
- Inter-process synchronization is needed to create and destroy shared memory segments and objects.
- GDB 6.8 allows multi-process debugging (forked and non-forked).
- Copy-on-Write and memory usage can be measured using smaps file parsing scripts.
- Inter-process synchronization should unlock if the process terminates unexpectedly (guarded lock is not enough). You can do it using:  
`http://isscvcs.cern.ch/cgi-bin/viewcvs-all.cgi/TriDAS/trigger/ts/toolbox/include/ts/toolbox/InterprocessMutex.h`
- The Linux kernel already has a cache mechanism in the "buffer cache" for read and write access. Measurements should take into account that.

## Presentations

M. Magrans de Abril, Shared Memory Prototype for LHC: Status and Plans [↗](#), CERN Multi-core R&D Effort Meeting, , 3rd July 2008.

---

-- MarcMagransDeAbril - 07 Jul 2008

- `factory20081002.zip`: Clean up version of the factory with tests

---

This topic: LCG > SharedMemoryLhc

Topic revision: r4 - 2011-03-17 - MarcMagransDeAbril



Copyright &© 2008-2021 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.  
or Ideas, requests, problems regarding TWiki? use [Discourse](#) or [Send feedback](#)