



IBM

# Building a Robust Linux Kernel piggybacking The Linux Test Project

---

**Subrata Modak & Balbir Singh,**  
*Linux Technology Center, IBM,  
India*

OLS 2008

*Dated 25/07/2008*

Linux is a registered trademark of Linus Torvalds.

IBM



# Agenda

- Introduction – some background checks
- Addressing some Criticism
- Breathing a new life into LTP
- Kernel code coverage statistics
- Role of LTP in testing Linux
- Early & Effective testing
- Simplest way to write an LTP test case
- Future Plans
- Conclusion



# Everybody is happy, and... they should be

- Linux Kernel is growing at a rapid rate and runs across numerous architectures...neighbor's envy – owner's pride

## Mission Critical...

- Keeping the kernel reliable, stable & robust

## Somebody doing the Job...??

- Yes, **The Linux Test Project**
- It was established to meet the very goals stated above



# Beware to ignore testing...!! you will end up in

- Frequent crashes
- Frequent Updates
- Unhappy users



# Introduction – some background checks

- First created by SGI in 2001 - brought organized and automated testing to Linux for first time
- Primary goal: provide a test suite that can validate the robustness, stability & reliability of Linux kernel
- A functional and regression test suite that allows to do stress testing as well
- No. of tests at kernel 2.3 was mere 100
- Today at 2.6.25 , it stands at 3000+ and growing...
- Tests numerous kernel features, namely syscalls, MM, IPC, I/O, device drivers, FS, Networking, etc
- What's the code written in ??... 95% in C, remaining in SHELL & PERL



# Addressing some Criticism

## People pointed out these gray areas to me

- LTP does not have
  - automatic kernel build, install, reboot and test
  - Good code coverage
  - parsible output/logs
  - Broken test cases

## My observation instead

- LTP was not designed to do auto build and test – *it was designed more of an handy regression test suite, testing all possible kernel APIs*
- Kernel code coverage cannot be drastically improved without corresponding test cases being made available – *we cannot enforce this, but can highlight the impact of not doing so*
- LTP logs/output are very neatly designed with proper tags – *analysis by human and programmes can be simple*
- Broken issues: some test cases were not cleaned for long – *LTP clearly distinguishes and documents meaning of keywords like PASS, FAIL, CONF, WARN, BROK, RETR & INFO*



# LTP's Output & Log samples

## SAMPLE LTP OUTPUT

```
<<<test_start>>>
tag=remap_file_pages01 stime=1208361993
cmdline="remap_file_pages01"
contacts=""
analysis=exit
initiation_status="ok"
<<<test_output>>>
remap_file_pages01
1 PASS : Non-Linear shm file OK

<<<execution_status>>>
duration=1 termination_type=exited
termination_id=0 corefile=no
cutime=7 cstime=2
<<<test_end>>>
```

## SAMPLE LTP LOG

```
Test Start Time: Wed Apr 16 21:47:41 2008
-----
Testcase          Result          Exit Value
-----          -
remap_file_pages01  PASS           0
faccessat01        FAIL           1
fallocate03        WARN           1
-----

Total Tests: 3
Total Failures: 1
Kernel Version: 2.6.18-53.1.13.el5
Machine Architecture: i686
Hostname: <sniff>
```



# Breathing a new life into LTP

- LTP started afresh from the early days of 2007 – developers put huge effort and added lot of fuel to testing Linux
- Numerous new test cases added, testing varied regions and kernel types – kdump, RT Linux, etc
- Massive cleanup to existing test cases done – applied 350 patches; adding 1000 sources, modifying 1000 equally and deleting around 250
- Numerous broken test case(s) issues were fixed



# List of test cases added – Jan07 till Mar08

Test cases Type	Total Sources (Initial Addition)	Avg. code size (bytes)	Contributors
Kdump,	26	2312	Sachin, Poornima, Jeff Burke, Cai Quan
Uts, Sysvipc, & Pid Namespace	27	2614	Rishikesh, Sergei, Suka & others
Inotify	4	5894	Andrew Vagin, Ricardo, Vivi Li & Others
Writev	7	7712	Breno Leitao
Swapon	4	8975	Ricardo & others
Numa	6	6986	Sivakumar, Pradeep & others
Remap_file_pages	3	6565	Ingo, Nick, Ricardo
Nfs Check Tests	1	1834	Kumar Gala
Posix_Fadvise & Fadvise64	5	4003	Masatake Yamato & others
Madvise	4	6572	Pavan Naregundi
Sendfile64	7	5625	Masatake Yamato
Arm Specific	1	1091	Riaz Ur Rahaman
Real Time Linux	101	3400	Nivedita, Sudhanshu, Chirag, Ankita, Darren, Sebastien, Gilles, Robert Schwebel & others
Fallocate	5	7071	Sharyathi Nagesh
Filescaps	11	2579	Sergei, Andrew Morgan & Henry Yei
Cpu Controllers	17	5134	Sudhir Kumar
Msgctl	12	7985	Nadia Derby
Ti-Rpc	588	3218	Aurélien Charbon



# Issues addressed in LTP-Refresh

- Release pattern revived to include results on various architectures
- Total 169 packages (265 Mb code) released, 31458 packages downloaded making avg. of 65 downloads/day
- GCOV kernel patches from linux-2.6.18 till 2.6.25
- Made RHEL5 LSPP EAL4+ certification test Suite available for IBM Hardwares
- SGI Common Criteria EAL4 certification test suite for RHEL5.1 on SGI Altix 4700 (ia64) and Altix XE (x86\_64) Systems was also made available



# Infrastructural improvements during LTP-Refresh

- Infrastructure gives the methodology/ways to run tests
- Adding discrete sequential run capability – run as many loops i want
- Auto mail back option of reports – bundle output/logs/failed tests and mail them back
- Generating default file for failed tests – this can be taken a command file for next ltp run
- Integrating better stress generation capability – allow more hogging of the CPU, Memory, I/O Channels, Storage & Network during test run
- Allow concurrent execution of test cases – test the SMP code



# LTP Output/Log (Report Generated on Mon Mar 31 12:14:29 CDT 2008)

<b>PASSED</b>	<b>FAILED</b>	<b>WARNING</b>	<b>BROKEN</b>	<b>RETIRED</b>	<b>CONFIG-ERROR</b>
---------------	---------------	----------------	---------------	----------------	---------------------

[Click Here for Detailed Report](#)  
[Click Here for Summary Report](#)

## Detailed Report

No	Test-Name	Command-Line	Test-Output	Termination-id
1	abort01	ulimit -c 1024;abort01	abort01 1 PASS : Test passed	0
93	faccessat01	faccessat01	faccessat01 6 FAIL : faccessat() Failed, errno=20 : Not a directory	1
94	fallocate01	fallocate01	fallocate01 0 WARN : System doesn't support execution of the test.	0
183	ftruncate04	ftruncate04	ftruncate04 1 CONF : The filesystem where /tmp is mounted does not support mandatory locks. Cannot run this test.	0

## Summary Report

<b>Test Summary</b>	<b>Pan reported some Tests FAIL</b>
LTP Version	LTP-20080331
Start Time	Mon Mar 31 12:14:29 CDT 2008
End Time	Mon Mar 31 13:05:21 CDT 2008
Log Result	<a href="/root/subrata/ltp/ltp-full-20080331/results">/root/subrata/ltp/ltp-full-20080331/results</a>
Output/Failed Result	<a href="/root/subrata/ltp/ltp-full-20080331/output">/root/subrata/ltp/ltp-full-20080331/output</a>
Total Tests	860
Total Failures	3
Kernel Version	2.6.21.3
Machine Architecture	i386
Hostname	sniff



# Kernel code coverage statistics

Kernel 2.6.24 & Dec 2007 LTP

Directory	Coverage	
fs	49.8%	10135/20367 lines
include/asm	49.4%	595/1204 lines
include/linux	58.7%	2239/3812 lines
include/net	56.2%	990/1762 lines
ipc	52.8%	1442/2729 lines
kernel	38.2%	9880/25837 lines
lib	42.2%	2105/4992 lines
mm	51.5%	6899/13396 lines
net	65.4%	630/964 lines
security	51.9%	666/1283 lines

Kernel 2.6.24 & March 2008 LTP

Directory	Coverage	
fs	52.9%	10778/20367 lines
include/asm	50.9%	613/1204 lines
include/linux	60.0%	2283/3812 lines
include/net	57.6%	1015/1762 lines
ipc	56.4%	1539/2729 lines
kernel	39.1%	10097/25837 lines
lib	43.2%	2159/4992 lines
mm	52.7%	7066/13396 lines
net	65.7%	633/964 lines
security	51.9%	666/1283 lines

## Code coverage Improvements

Subsystem	% Increase
Filesystems	3.1
include/asm	1.5
include/linux	1.3
include/net	1.4
ipc	3.6
kernel	0.9
lib	1.0
mm	1.2
net	0.3
security	0

\*\* Test cases excluded from code coverage: **Kdump, RT, Dots, Open Posix, Open HPI, Pounder21 & SE Linux**



# Analysis of code coverage

- Coverage better between 2 runs – indicative of LTP's progress
- Subsystems *fs* & *include/asm* is now above 50%
- Some more interesting facts
  - LTP needs to do a better job covering error paths – we propose **LTP-robust** subproject to work well with fault injection
  - Not possible to test every config/boot option & extract coverage
  - Not possible to handle code coverage not exposed to user space – machine configured with SPARSEMEM, FLATMEM, DISCONTIGMEM
  - Several areas/subsystems does not have any code coverage – we need to write test cases for them as well



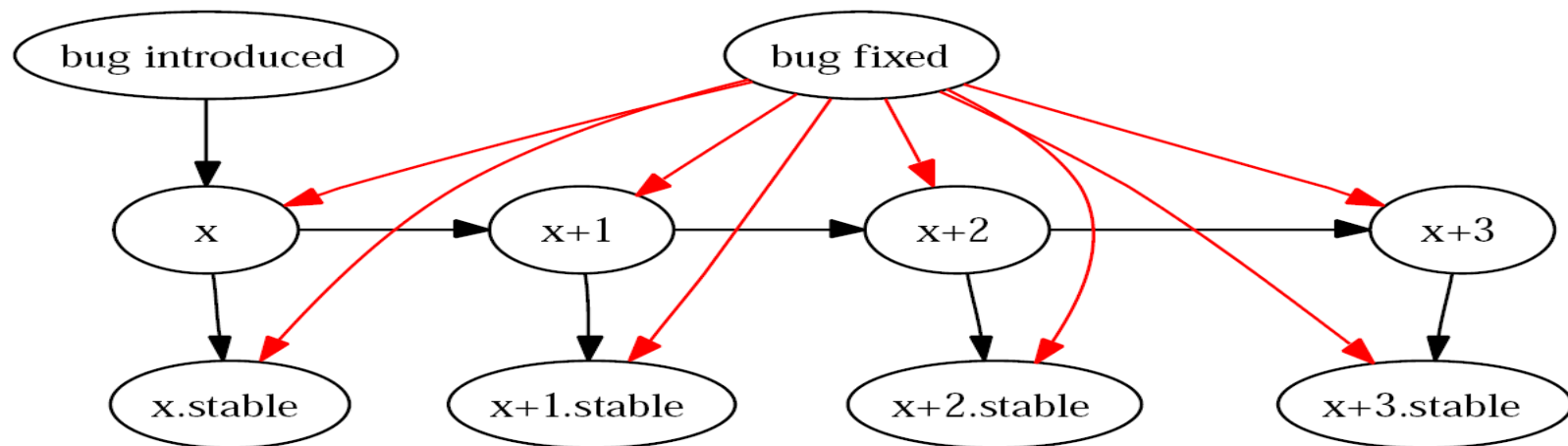
# Role of LTP in testing Linux

- Broader software testing categories includes – compilation, unit, functional, system, stress & performance testing
- Unit testing – execution for individual/isolated feature alone,
- Functional/regression testing through comparison of successive kernel releases
- System testing – "C" Library and user interfaces provided by the kernel
- Stress testing – through specific test cases, additional background noise, concurrent test execution
- Unable to do compilation & performance testing



# Early & Effective testing

- Spend more time on design – else be ready to spend 40 to 1000 times more time in fixing code after deployment
- Testing cannot catch bug during requirement analysis, but can definitely do before code deployment



Hypothetical Example: Sample bug fix flow for a bug introduced in version x and fixed in version x+3

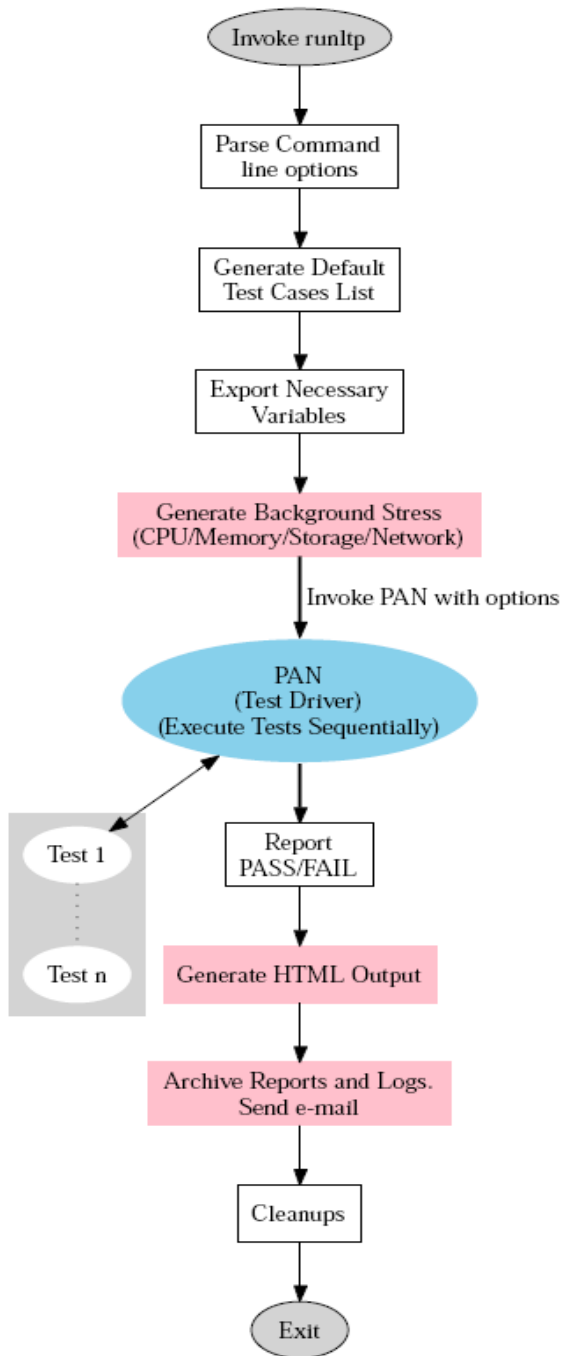


# Simplest way to write an LTP test case

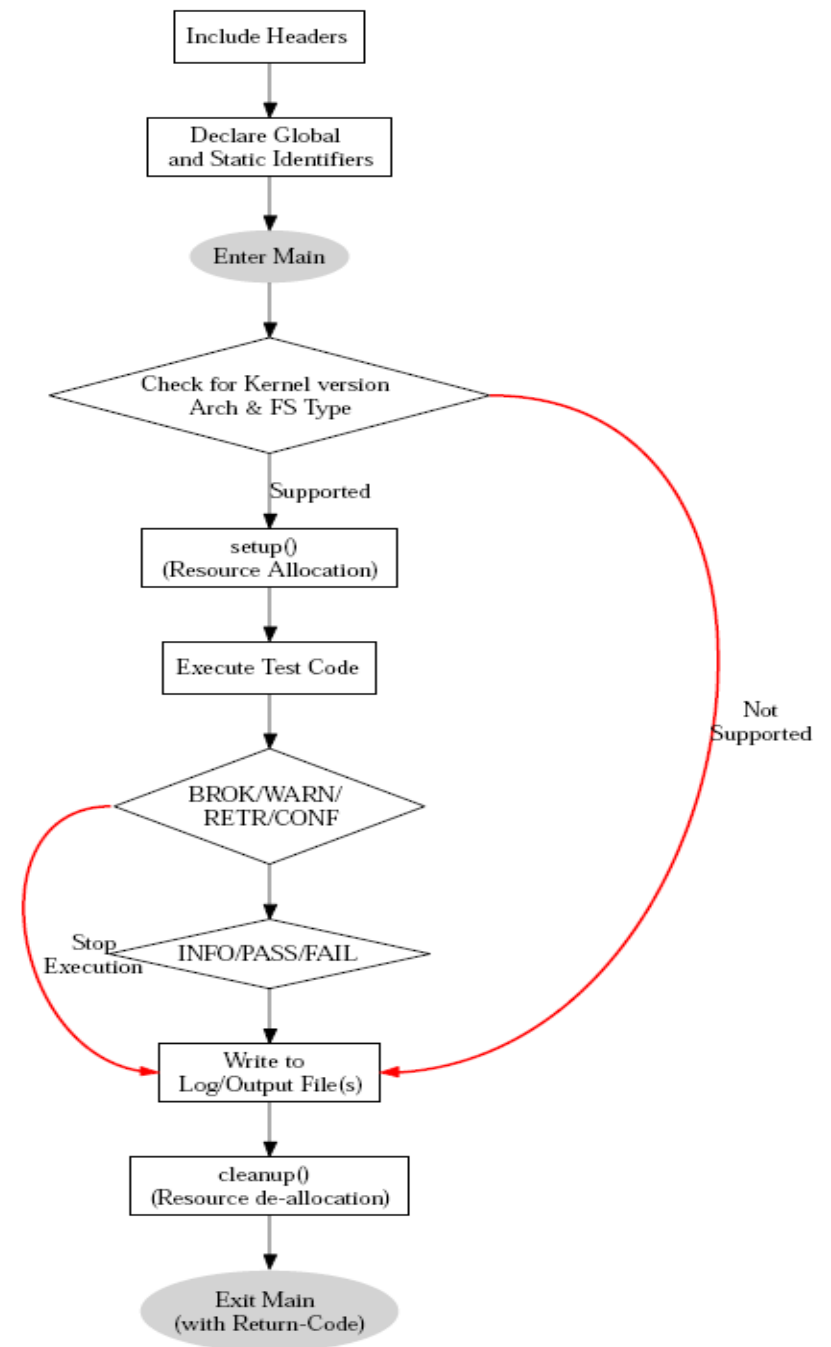
- This Paper will not focus in intricate details of writing an LTP test case
- We instead focus on presenting a set of work flows
- Ways to write test cases can be found from already published papers
  - [Testing Linux with Linux Test Project, Paul Larson, OLS 2002](#)
  - [Improving the Linux Test Project with Kernel Code Coverage Analysis, Paul Larson, OLS 2003](#)



# LTP Suite execution framework



# Individual test case execution framework



# Future Plans

- LTP-devel rpms – a light-weight package containing LTP-specific headers, libraries, man pages – enabling developers to write LTP unit test cases
- LTP-*mm* tree: contribute test cases to LTP-*mm* tree when your feature is in kernel-*mm* tree, or, you have just submitted your patch to LKML
- Enabling test case submission in any programming language which returns just 0/1 on pass/fail
- LTP aims to include test case in the areas of Power Management, Controllers/Containers, KDUMP, Union Mount, Shared subtree, missing syscalls, bleeding edge kernels, etc
- Enhancement in LTP Infrastructure
  - Development of XML logs/output
  - .config based build & execution
  - Network based installation, execution & report collection
- Entire bunch of test cases to be made concurrency safe
- Efforts on functional & regression testing to strengthen; also to add benchmark infrastructure in the long run



# Conclusion

- LTP is Open Source: everybody can say how it should move forward, what it should address and... avoid !!
- LTP community highly appreciates patches in any form – *the benefit of which directly goes to the community*
- It will continue to be as a major functional & regression suite, and will keep growing along with the unstoppable kernel

For all these to happen, LTP will require:

- 1) More active participation from Kernel Developers, and
- 2) Tons & tons of test cases from you all



# Acknowledgments

- Robert Williamson, IBM, for his inputs to and review of the draft of this paper
- Premalatha Nair & Sudarshan Rao (my Managers) & my team mates for their active support, encouragement and enthusiasm
- A special thanks to all developers whose immense contribution keeps this project growing



# Legal Statement

- International Business Machines Corporation 2008. Permission to redistribute in accordance with Linux Symposium submission guidelines is granted; all other rights reserved.
- This work represents the view of the authors and does not necessarily represent the view of IBM.
- IBM, IBM logo, ibm.com, and WebSphere, are trademarks of International Business Machines Corporation in the United States, other countries, or both.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Other company, product, and service names may be trademarks or service marks of others.
- References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.
- INTERNATIONAL BUSINESSMACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.



# References

- [1] Linux test project download page. [https://sourceforge.net/project/showfiles.php?group\\_id=3382](https://sourceforge.net/project/showfiles.php?group_id=3382).
- [2] Linux test project home page. <https://sourceforge.net/projects/ltp>.
- [3] Linux test project mailing list. [https://sourceforge.net/mailarchive/forum.php?forum\\_name=ltp-list](https://sourceforge.net/mailarchive/forum.php?forum_name=ltp-list).
- [4] Linux test project source code repository. <http://ltp.cvs.sourceforge.net/ltp/>.
- [5] B. Boehm. Software Engineering Economics. Prentice Hall, 1981.
- [6] N. Hinds. Kernel korner: The linux test project: Finding 500 bugs in 50 different kernel versions is the fruit of this thorough linux testing and code coverage project. Linux Journal, 2004. <http://www.linuxjournal.com/article/7445>.
- [7] M. Iyer. Linux test project documentation howto. <http://ltp.sourceforge.net/documentation/how-to/ltp.php>.
- [8] P. Larson. Improving the linux test project with kernel code coverage analysis. Linux Symposium 2003.
- [9] P. Larson. Testing linux with linux test project. Linux Symposium 2002.
- [10] P. Larson, R. Williamson, and M. Ridgeway. Linux test project technical papers. <http://ltp.sourceforge.net/documentation/technicalpapers>.



■ ■ ■ **Q**uestions ■ ■ ■

Mail your **S**uggestions to:

*Subrata Modak <subrata@linux.vnet.ibm.com>*,

*Balbir Singh <balbir@linux.vnet.ibm.com>*

- <https://sourceforge.net/projects/ltp>
- <http://ltp.sourceforge.net/>

IBM

