

Linux Test Project

활용 가이드

2009년 12월 15일

정보통신산업진흥원

목 차

1. LTP Test Suite 소개	1
2. LSB Test Suite의 다운로드	1
3. LTP Test Suite의 콘텐츠	2
4. LTP Test Suite의 컴파일링	4
5. LTP Test Suite의 실행	5
가. 테스트의 기본 설정 실행	7
나. 테스트의 유저 설정 실행	8
6. LTP Test Suite의 실행결과 확인	9
가. 화면 출력	9
나. 로그파일 출력	9

1. LPT Test Suite 소개

- LTP(Linux Test Project)는 리눅스의 신뢰성, 호환성, 안정성을 입증하여 오픈소스 커뮤니티에 Test Suite를 제공하는 오픈소스 프로젝트이다.
- LTP Test Suite는 리눅스 운영체제를 다양한 관점에서 테스트 할 수 있도록 개발된 자동, 반자동 Test Suite들의 집합체이다.
- LTP를 구성하는 프로그래밍언어는 ANSI-C(94%), Bash(5%), Perl(0.62%)이다.
- 본 문서에서는 LTP Test Suite의 다운로드, 컴파일링, 실행, 결과 출력에 대한 전반적인 내용을 설명한다.

2. LPT Test Suite의 다운로드

- LTP Test Suite 소스의 안정 버전은 기본적으로 매월 Sourceforge.net에 `ltip-intermediate-yyyymmdd.tgz` 형식으로 릴리즈 된다. Test Suite는 리눅스 커널의 안정성을 입증하고 개발SW를 릴리즈 하는데 사용되며, Source와 RPM 형식으로 제공된다.
- 패키지 `ltip-intermediate-yyyymmdd.tgz`는 LTP 웹사이트나 `wget` 커맨드를 사용하여 다운로드 할 수 있다.
- 통합 다운로드 사이트
 - <http://sourceforge.net/projects/ltip/files/>
- `wget`을 사용한 소스파일 다운로드 예
 - `wget http://downloads.sourceforge.net/project/ltip/LTP%20Source/ltip-20090831/ltip-intermediate-20090921.tgz?use_mirror=ufpr`
- 다음의 익명 CVS를 통해서 LTP Test Suite 소스의 최신 CVS 스냅샷을 다운로드 할 수 있다.
 - `cvs -d:pserver:anonymous@ltip.cvs.sourceforge.net:/cvsroot/ltip login`
 - `cvs -z3 -d:pserver:anonymous@ltip.cvs.sourceforge.net:/cvsroot/ltip co ltp`

- 각 하위 디렉터리에 생성된 것들을 제외하고 CVS를 다운로드 하려면 아래의 명령어를 사용한다.
 - `cvs -d:pserver:anonymous@ltp.cvs.sourceforge.net:/cvsroot/ltp login`
 - `cvs -z3 -d:pserver:anonymous@ltp.cvs.sourceforge.net:/cvsroot/ltp export -D "MONTH DAY, YEAR HH:MMam/pm" ltp`

3. LPT Test Suite의 콘텐츠

- LTP 루트 디렉터리 /ltp-intermediate-yyyymmdd 에는 셸 스크립트와 다양한 소스 디렉터리들이 포함되어 있다. 이 스크립트들은 테스트 케이스를 수행하는데 사용된다. 스크립트는 pan이라 불리는 테스트 드라이버를 사용하는 테스트 케이스를 실행한다.

가. 스크립트

1) IDcheck.sh

Test Suite는 시스템에 생성된 유저와 그룹을 요구한다. 이 스크립트는 유저와 그룹을 생성하며, LTP 루트 디렉터리에 위치한 최상위 Makefile을 불러온다. 또한, 루트 유저로 실행하여야 한다.

2) runltp

본 드라이버 스크립트는 LTP에서 중요한 테스트를 수행한다. 필요 시 테스트의 사용자가 지정한 테스트를 수행하기 위해 쉽게 수정할 수 있다. 기본적으로 이 스크립트는 다음의 테스트를 수행한다.

- filesystem stress tests
- disk I/O tests
- memory management stress tests
- ipc stress
- scheduler tests
- commands functional varification tests
- system call functional varification tests

3) runalltest.sh

본 스크립트는 기본적으로 모든 테스트를 수행한다. 그러나 일부 수행하지 않는 테스트가 있다.

4) networktests.sh

네트워킹 연결 테스트를 수행한다. 이 스크립트를 수행하기 전에 두 개의 환경 변수인 RHOST(Remote Host Name)와 PASSWD(원격 HOST의 루트 패스워드)를 설정해야 한다.

5) diskio.sh

플로피 드라이브와 CD-ROM 드라이브의 연결 테스트를 수행한다. 이 테스트를 수행하기 위해서는 각 드라이브의 매체(플로피 디스크, CD-ROM 미디어)가 필요하다.

나. 디렉터리 설명

1) testcases

모든 테스트 케이스의 소스 코드가 포함되어 있다. 테스트 케이스는 서브시스템이나 서비스가 계획된 이후에 이름 지어진 디렉터리 하위에 위치한다. 예를 들어, 메모리 관리 테스트는 testcases/mem 등의 하위에 위치한다.

2) runtest

테스트 수행을 위한 pan에 의해 사용되어 지는 커맨드 파일들이 위치한다. 각 파일에는 테스트를 수행하기 위해 실행해야만 하는 인수인 테스트 이름과 커맨드의 이름이 포함되어 있다. 예를 들어, syscalls 파일은 아래 예제1 과 같은 입력들을 포함한다.

예제1. Command File의 내부 구조

# Name of test	Command
access01	access01
alarm04	cp -f \$LTPROOT/test-cases/bin/sig_rev \$TMP;alarm04
chdir01A	symlink01 -T chdir01; chdir01

3) lib

LTP-harness API's (libltp.a)를 위한 소스코드가 위치한다. libltp.a 라이브러리는 로그 테스트 결과와 상태를 위해 이 API 함수들이 사용하는 모든 테스트 케이스와의 링크를 제공한다.

4) include

LTP API-harness 함수를 사용하는 모든 테스트 케이스에 포함된 HEADER 파일이 위치한다.

5) testcases/bin

컴파일링 되어 설치된 실행 가능한 테스트 케이스가 위치한다.

4. LPT Test Suite의 컴파일링

- Test Suite를 컴파일 하기 위해서는 LTP Test Suite의 루트 디렉터리인 /ltp-intermediate-yyyymmdd로 디렉터리를 이동하고 configure를 입력한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# ./configure
```

- 컴파일을 진행하기 위해서는 gcc와 lex 패키지가 필요하다. 해당 패키지가 설치되어 있지 않다면 다음을 수행하여 설치를 진행한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# yum install gcc flex
```

- 컴파일이 완료되었으면 이제 Test Suite를 설치하기 위해 make와 make install을 순서대로 실행한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# make
```

```
[root@localhost ltp-intermediate-yyyymmdd]# make install
```

- 설치하는 루트 유저로 수행해야한다. 이 작업의 수행은 /ltp-intermediate-yyyymmdd/test-cases/bin 디렉터리에 해당 파일을 실행시키는 Symbolic 링크를 생성한다.
- 테스트 스크립트(예:runltp)는 커맨드 파일 안에 열거된 테스트를 기반으로 이 디렉터리에 제공된 바이너리를 실행한다.

5. LPT Test Suite의 실행

□ 일반적으로 LPT Test Suite에는 테스트 케이스를 실행하기 위해 사용하는 3개의 스크립트(runltp, networktests.sh, diskio.sh)가 있다. 가장 대표적으로 사용하는 스크립트는 runltp 이며, [표1]과 같은 옵션을 제공한다. 만약 옵션 없이 실행한다면 아래처럼 기본적인 테스트만을 수행한다.

- 기본 테스트 설정의 실행
- 화면에 테스트에 대한 자세한 정보를 출력
- 로그파일이나 테스트 결과를 생성하지 않음

[표1. LTP 테스트 슈트 옵션]

옵션	기능
-a EMAIL_TO	입력한 E-mail 주소로 모든 리포트를 전송
-c NUM_PROCS	부가적인 백그라운드 CPU 부하 아래에서 LTP를 실행 [NUM_PROCS = no. of processes creating the CPU Load by spinning over sqrt() (Defaults to 1 when value)]
-C FAILCMDFILE	모든 실패한 테스트 케이스를 포함한 커맨드 파일을 지정
-d TMPDIR	임시파일이 생성될 디렉터리 지정
-D NUM_PROCS,NUM_FILES,NUM_BYTES, CLEAN_FLAG	세컨드 스토리지의 부가적인 백그라운드 부하 아래에서 LTP를 실행 (구분자:',') [NUM_PROCS = no. of processes creating Storage Load by spinning over write()] [NUM_FILES = Write() to these many files (Defaults to 1 when value 0 or undefined)] [NUM_BYTES = write these many bytes (defaults to 1GB, when value 0 or undefined)] [CLEAN_FLAG = unlink file to which random data written, when value 1]
-e	현재 LTP release의 날짜를 출력
-f CMDFILES	유저가 선언한 테스트 케이스의 리스트를 실행(구분자:',')
-g HTMLFILE	추가된 HTML 출력 포맷을 생성
-h	도움말 출력(가능한 모든 옵션을 출력)

옵션	기능
-i NUM_PROCS	IO Bus위의 추가적인 백그라운드 부하 아래에서 LTP를 실행 [NUM_PROCS = no. of processes creating IO Bus Load by spinning over sync()]
-l LOGFILE	테스트 결과를 로그파일로 생성
-m NUM_PROCS,CHUNKS,BYTES,HANGUP_FLAG	주 메모리 위의 추가적인 백그라운드 부하 아래에서 LTP를 실행(구분자:',') [NUM_PROCS = no. of processes creating main Memory Load by spinning over malloc()] [CHUNKS = malloc these many chunks (default is 1 when value 0 or undefined)] [BYTES = malloc CHUNKS of BYTES bytes (default is 256MB when value 0 or undefined)] [HANGUP_FLAG = hang in a sleep loop after memory allocated, when value 1]
-N	모든 네트워크 테스트를 수행
-n	백그라운드 네트워크 트래픽과 함께 LTP를 수행
-o OUTPUTFILE	파일로 출력
-p	ASCII 포맷으로 로그파일을 생성
-q	화면에 최소한의 정보만 출력
-r LTPROOT	LTP 루트 디렉터리의 경로를 지정
-s PATTERN	PATTERN과 매치되는 테스트 케이스만을 수행
-t DURATION	시간설정 ex) -t 60s = 60 초 -t 45m = 45 분 -t 24h = 24 시간 -t 2d = 2 일
-T REPETITION	Test Suite를 반복해서 수행
-v	테스트 케이스의 유저 리스트를 얻기 위해 wget을 사용
-w CMDFILEADDR	이 Test Suite의 다양한 인스턴스를 수행

가. 테스트의 기본 설정 실행

- LTP의 루트 디렉토리로 이동 후 runltp 스크립트를 실행한다.

```
[root@localhost ~]# cd ltp-intermediate-yyyymmdd
```

```
[root@localhost ltp-intermediate-yyyymmdd]# ./runltp
```

- 원하는 옵션이 있을 경우 상단의 옵션 리스트를 참고하여 옵션을 추가한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# ./runltp -p -l logfile_name.log
```

예제2. 결과 로그 파일의 내용

```
Test Start Time: Mon Sep 28 13:38:45 2009
```

```
-----
Testcase          Result      Exit Value
-----
abort01           PASS        0
accept01          PASS        0
accept4_01        PASS        0
access01          PASS        0
access02          PASS        0
access03          PASS        0
```

```
... 중략 ...
```

```
mkdir_tests01    PASS        0
mv_tests01       PASS        0
size01           PASS        0
smt_smp_enabled  PASS        0
smt_smp_affinity PASS        0
ht_interrupt     PASS        0
```

```
-----
Total Tests: 1225
Total Failures: 75
Kernel Version: 2.6.18-128.GX2
Machine Architecture: x86_64
Hostname: skcc_oss
```

나. 테스트의 유저 설정 실행

- 테스트 케이스의 설정을 변경하여 테스트를 수행하고자 할 경우에는 다음의 내용을 수행한다.

1) 커맨드 파일을 생성한다.

커맨드 파일은 인수를 포함한 테스트 케이스와 테스트 케이스 Tag를 포함해야 한다.

예제3. Command File의 내부 구조

```
#Tag      Test case
#-----
mtest01    mtest01 -p 10
mmstress   mmstress -x 100
fork01     fork01
chdir01    symlink01 -T chdir01
#-----
```

2) LTP 루트 디렉터리인 /ltp-intermediate-yyyymmdd 디렉터리로 이동한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# yum install gcc flex
```

3) 이 테스트를 수행하기 위한 커맨드 파일 옵션인 -f를 사용하여 runltp를 실행한다.

```
[root@localhost ltp-intermediate-yyyymmdd]# ./runltp -p -l logfile_name.log -f
my_command_file
```

6. LPT Test Suite의 실행결과 확인

가. 화면 출력

- 테스트 케이스, 테스트 드라이버, 스크립트는 다양한 정보 메시지를 화면에 출력한다.
- 테스트 케이스는 정보를 프린트하기 위해 USCTEST 함수를 사용한다.
- 테스트 드라이버 pan은 또 다른 테스트로부터 별도의 출력인 tags를 프린트 한다.
- 옵션에 따라, 스크립트를 실행하는 동안 이 메시지들의 내용을 조절할 수 있다.

나. 로그파일 출력

- 테스트 드라이버 pan은 테스트의 성공 또는 실패를 결정하기위해 테스트 케이스의 종료 값을 사용한다. 만약 테스트 케이스가 zero pan 값으로 종료될 경우 PASS로 기록하고 zero pan이 아닐 경우 FAIL로 기록한다.
- 테스트 드라이버 pan은 로그 파일을 생성한다. 테스트 케이스의 실행으로부터 특정 결과인 이 로그파일은 로그 결과를 위하여 runalltest.sh 스크립트에 -l 로그파일_이름 옵션을 사용할 수 있다. 로그파일은 기본적으로 results 디렉터리에 생성되며, 결과 로그파일은 2개의 포맷으로 제공된다.

1) ASCII 포맷 : -p 옵션

예제4. ASCII 포맷의 결과 화면

```
Test Start Time: Mon Sep 28 11:06:37 2009
-----
Testcase          Result      Exit Value
-----
mtest01           FAIL        1
mmstress          PASS        0
-----
Total Tests: 2
Total Failures: 1
Kernel Version: 2.4.19
Machine Architecture: i686
Hostname: lazy
```

2) SCRIPTS 포맷 : 기본 값

기본 포맷 방식이며 테스트 결과 분석 보고서를 생성하는 스크립트에 의해 사용될 수 있다.

예제5. SCRIPTS 포맷의 결과 화면

```
tag=mtest01 stime=1043947511 dur=0 exit=exited stat=1 core=no cu=0 cs=11
tag=mmstress stime=1043947511 dur=41 exit=exited stat=0 core=no cu=671
cs=3383
tag=fork01 stime=1043947552 dur=0 exit=exited stat=0 core=no cu=0 cs=1
tag=chdir01A stime=1043947552 dur=0 exit=exited stat=0 core=no cu=0 cs=0
```