Installing SPEC CPU2006 Under Unix, Linux, and Mac OS X

Last updated: $Date: 2014-11-21 16:44:34 -0500 (Fri, 21 Nov 2014) $ by $Author: CathySandifer $
(To check for possible updates to this document, please see http://www.spec.org/cpu2006/Docs/)

This document has been updated for SPEC CPU2006 V1.2, including a new example installation.

Contents

Note: links to SPEC CPU2006 documents on this web page assume that you are reading the page from a directory that also contains the other SPEC CPU2006 documents. If by some chance you are reading this web page from a location where the links do not work, try accessing the referenced documents at one of the following locations:

- www.spec.org/cpu2006/Docs/
- The `$SPEC/Docs/` (Unix) or `%SPEC%\Docs\` (Windows) directory on a system where SPEC CPU2006 has been installed.
- The `Docs/` directory on your SPEC CPU2006 distribution DVD.

Installation Steps

The SPEC CPU2006 suite has been tested under Unix, Linux, Mac OS X and Windows XP systems. Your DVD can be installed under many operating systems.

Reminder: the SPEC license allows you to install on multiple systems as you may wish within your institution; but you may not share the software with the public.

The installation procedure for Unix, Linux, and Mac OS X is as follows:

1. Review Pre-requisites

Review the hardware and software requirements, in system-requirements.html

2. Create destination. Have enough space, avoid space.

Create a directory on the destination disk. You should make sure that you have a disk that has at least 8GB free. (For more information on disk usage, see system-requirements.html .)

Don't put spaces in the path: even if you make it through the installation (doubtful), you are just asking for trouble, because there may be multiple programs from both SPEC and from your compiler that expect space to be an argument delimiter, not part of a path name. (This being the *Unix* install guide, you wouldn't have thought of using spaces in in the first place, would you?)

3. Mount the DVD

Insert the the DVD, and, if necessary, issue a `mount` command for it. For many operating systems, the DVD will be automatically mounted. If not, you may have to enter an explicit `mount` command. If your operating system supports the Rock Ridge Interchange Protocol extensions to ISO 9660, be sure to select them, unless they are the default. The following examples are not intended to be comprehensive, but may get you started or at least give you clues which manpages to read:

AIX:	`mount -v cdrfs -r /dev/cd0 /cdrom`
HP-UX:	`mount -o cdcase /dev/disk/disk5 /mnt/cdrom/`
Linux:	`mount -t iso9660 -o ro,exec /dev/cdrom /mnt`

Solaris:	If Volume Management is running, you should find that the DVD is automatically mounted, as `/cdrom/label_of_volume/` If not, you should be able to mount it with commands similar to this:
	`mkdir /mnt1`
	`mount -F hsfs -o ro /dev/dsk/c0t6d0s0 /mnt1`

Virtual Machines	If you are running in a virtual machine, you will need to convince the host operating system to allow your guest OS to have access to the DVD. The means of accomplishing this will vary. For reference, the following worked with a Linux guest running under Virtual Box V4.0.6, with Windows 7 as the host: (1) Shut down the virtual machine (don't just pause it; tell it to run its shutdown procedure). (2) The Settings dialog should now be visible (it's grayed out if the machine state is not shut down). (3) Use Settings to configure the DVD drive as both available to the guest OS and as "passthrough". (4) Boot the virtual machine. (5) Log in. (6) Insert the DVD. (7) At this point, the DVD was automatically mounted as `/media/SPEC_CPU2006`.

Note that you may need root privileges to mount the DVD.

The following paragraphs assume that your DVD drive is on the same system as where you wish to install. If it is on a different system, please see Appendix 1.

4. Set your directory to the DVD

If you haven't already done so by now, start a Terminal window (aka "command window", "shell", "console", "terminal emulator", "character cell window", "xterm", etc.) and issue a `cd` command to set your current working directory to the directory where the DVD is mounted. The exact command will vary depending on the label on the media, the operating system, and the devices configured. It might look something like one of these:

```
$ cd /Volumes/SPEC_CPU2006
$ cd /media/SPEC_CPU2006
$ cd /dvdrom/spec_cpu2006
$ cd /mnt
```

5. Use install.sh

Type:

```
./install.sh
```

q. Do you have to be root? Occasionally, users of Unix systems have asked whether it is necessary to elevate privileges, or to become 'root', prior to entering the above command. SPEC recommends (*) that you do not become root, because: (1) To the best of SPEC's knowledge, no component of SPEC CPU needs to modify system directories, nor does any component need to call privileged system interfaces.
(2) Therefore, if you find that it appears that there is some reason why you need to be root, the cause is likely to be outside the SPEC toolset - for example, disk protections, or quota limits. (3) For safe benchmarking, it is better to avoid being root, for the same reason that it is a good idea to wear seat belts in a car: accidents happen, humans make mistakes. For example, if you accidentally type:

```
kill 1
```

when you meant to say:

```
kill %1
```

then you will very grateful if you are not privileged at that moment.

(*) This is only a recommendation, not a requirement nor a rule.

5.a. Destination selection

Depending on your installation type, you may be prompted for a destination directory:

```
SPEC CPU2006 Installation
Top of the CPU2006 tree is '/Volumes/SPEC_CPU2006'
Enter the directory you wish to install to (e.g. /usr/cpu2006)
/Users/kgoel/cpu2006
```

When answering the above question, note that you will have to use syntax acceptable to sh (so you might need to say something like "$HOME/mydir" instead of "~/mydir"). As mentioned above, don't use spaces.

Note: You can also specify the destination directory in the command line, using the -d flag, for example, like this:

```
./install.sh -d /Users/kgoel/cpu2006
```

The installation procedure will show you the directories that will be used to install from and to. You will see a message such as this one:

```
Installing FROM /Volumes/SPEC_CPU2006
Installing TO /Users/kgoel/cpu2006

Is this correct? (Please enter 'yes' or 'no')
yes
```

Enter "yes" if the directories match your expectations. If there is an error, enter "no", and the procedure will exit, and you can try again, possibly using the -d flag mentioned in the note above.

5.b. Toolset selection

The installation procedure will attempt to automatically determine your current platform type (hardware architecture, operating system, etc.) In some cases, the tools may identify several candidate matches for your architecture.

You typically do not have to worry about whether the toolset is an exact match to your current environment, because the toolset selection **does not affect your benchmark scores**, and because the installation procedure does a series of tests to ensure that the selected tools work on your system.

Examples: (1) the installation procedure may determine that SPEC tools built on version "N" of your operating system are entirely functional on version "N+3". (2) Tools built on one Linux distribution often work correctly on another: notably, certain versions of SuSE are compatible, from a tools point of view, with certain versions of RedHat. (3) Tools built on AMD chips with 64-bit instructions ("amd64") are compatible with Intel chips that implement the same instruction set under the names "EM64T" or "Intel 64" (but not compatible with chips that implement the Itanium instruction set, abbreviated "ia64"). (4) Often, though not always, 32-bit toolsets work correctly on 64-bit operating systems.

Mostly, you don't need to worry about all this, because the installation procedure does a comprehensive set of tests to verify compatibility.

If at least one candidate match is found, you will see a message such as:

```
The following toolset is expected to work on your platform.  If the
automatically installed one does not work, please re-run install.sh and
exclude that toolset using the '-e' switch.

The toolset selected will not affect your benchmark scores.

macosx                          For MacOS X 10.4+ on Intel systems.
                                Built on MacOS X 10.6.6 with GCC 4.0.1, using
                                the 10.4u SDK.
```

If the installation procedure is unable to determine your system architecture, you will see a message such as:

```
We do not appear to have vendor supplied binaries for your
architecture.  You will have to compile the tool binaries by
yourself.  Please read the file

    /Volumes/SPEC_CPU2006/Docs/tools_build.html

for instructions on how you might be able to build them.
```

If you see that message, please stop here, and examine the file tools-build.html .

Note: If the tools that are automatically installed on your system do not work, but you know that another set of tools that is in the list will work, you can exclude the ones that do not work. You may be instructed to do this during the first installation. Use the **-e** flag for install.sh, for example:

```
./install.sh -e linux-redhat72-ia32
```

The above will cause the tools for `linux-redhat72-ia32` to be excluded from consideration.

Alternatively, you can explicitly direct which toolset is to be used with the `-u` flag for install.sh, for example:

```
./install.sh -u linux-suse10-amd64
```

The above will cause the tools for `linux-suse10-amd64` to be installed, even if another toolset would have been chosen automatically. If you specify tools that do not work on your system, the installation procedure will stop without installing any tools.

5.c. The files are unpacked and tested

Thousands of files will be unpacked from the distribution media, and quietly installed on your destination disk. (If you would prefer to see them all named you can set `VERBOSE=1` in your environment before installing the kit.) Various tests will be performed to verify that the files have been correctly installed, and that the tools work correctly. You should see summary messages such as these:

```
=============================================================
Attempting to install the the macosx toolset... <<-- or whatever toolset was selected

Unpacking CPU2006 base files (129.8 MB)
Unpacking 400.perlbench benchmark and data files (61.5 MB)
.
.       (lines omitted)
.
Checking the integrity of your source tree...


Checksums are all okay.

Unpacking binary tools for macosx...        <<-- your toolset

Checking the integrity of your binary tools...

Checksums are all okay.
Testing the tools installation (this may take a minute)

.....................................................................o.............o.................
.......o.........................o.....................................................................

Installation successful.  Source the shrc or cshrc in
/Users/kgoel/cpu2006                        <<-- your directory
to set up your environment for the benchmark.
```

At this point, you will have consumed about 1.5GB of disk space on the destination drive.

6. Source shrc or cshrc

Change your current directory to the top-level SPEC directory and source either `shrc` or `cshrc`:

- For example, if you are using a Bourne-compatible shell (such as ash, bash, ksh, zsh), you could type:

```
Users$ cd /Users/kgoel/cpu2006/
cpu2006$ . ./shrc              <-- that's dot-space-dot-slash-shrc
```

- If you are using a csh-compatible shell, you could type:

```
Users% cd /Users/kgoel/cpu2006
cpu2006% source cshrc
```

(History: The cshrc script was added in SPEC CPU2006 V1.0.)

The effect of the above commands is to set up environment variables and paths for SPEC.

From this point forward, we are testing basic abilities of the SPEC CPU2006 kit, including compiling benchmarks and running them. You may skip the remaining steps if all of the following are true:

1. You are confident that the previous steps have gone smoothly.
2. You will not be compiling the benchmarks.
3. Someone else has given you pre-compiled binaries.

Warning: even if someone else supplies binaries, you remain responsible for compliance with SPEC's Fair Use rule and the CPU2006 run rules .

---

7. Try to build one benchmark

Change to the `config` directory, and test that you can build a benchmark using a config file supplied for your system. For example:

```
$ cd $SPEC/config
$ cp Example-macosx-gcc421.cfg Khushboo-macosx.cfg
$ runspec --config=Khushboo-macosx.cfg --action=build --tune=base bzip2
```

The above command assumes that you can identify a config file (in the directory `$SPEC/config`) that is appropriate for you. In this case, the user started with `Example-macosx-gcc421.cfg`. Your starting point will probably differ; here are some resources to help:

- Hints about picking Linux config files are included in the long example below.
- For a quick introduction to config files, see "About Config Files" in runspec.html
- You may need to copy one of the supplied config files, and edit it for your system.
- For example, you may need to edit it to have the correct path to your compiler!

The "`--tune=base`" above indicates that we want to use only the simple tuning, if more than one kind of tuning is supplied in the config file.

8. Try running one benchmark with the test dataset

Test that you can run a benchmark, using the minimal input set - the "test" workload. For example:

```
$ runspec --config=Khushboo-macosx.cfg --size=test --noreportable --tune=base \
--iterations=1 bzip2
```

The "\" above indicates that the command is continued on the next line. The "`--noreportable`" ensures that the tools will allow us to run just a single benchmark instead of the whole suite, and "`--iterations=1`" says just run the benchmark once.

Check the results in `$SPEC/result`

9. Try a real dataset

Test that you can run a benchmark using the real input set - the "reference" workload. For example:

```
$ runspec --config=Khushboo-macosx.cfg --size=ref \
     --noreportable --tune=base --iterations=1 bzip2
```

Check the results in `$SPEC/result`.

10. Try a full (reportable) run

If everything has worked up to this point, you may wish to start a full run, perhaps leaving your computer to run overnight. The extended test will demand significant resources from your machine, including computational power and memory of several types. In order to avoid surprises, before starting the reportable run, you should review the section About Resources , in system-requirements.html.

Have a look at runspec.html to learn how to do a full run of the suite.

The command `runspec -h` will give you a brief summary of the many options for `runspec`.

To run a reportable run of the integer suite with simple (baseline) tuning:

```
C:\cpu2006> runspec --tune=base --config=Khushboo-macosx.cfg int
```

Example Installation

Here is a complete Linux installation, with interspersed commentary. This example follows the steps listed above. We assume that Steps 1 through 3 are already complete (the pre-requisites are met, we have enough space, the DVD is mounted).

Step 4: Set the current working directory to the DVD:

```
$ cd /media/SPEC_CPU2006
```

Step 5: Invoke install.sh. When prompted, we enter the destination directory:

```
$ ./install.sh

SPEC CPU2006 Installation

Top of the CPU2006 tree is '/media/SPEC_CPU2006'
Enter the directory you wish to install to (e.g. /usr/cpu2006)
/cpu2006

Installing FROM /media/SPEC_CPU2006
Installing TO /cpu2006

Is this correct? (Please enter 'yes' or 'no')
yes

The following toolset is expected to work on your platform.  If the
automatically installed one does not work, please re-run install.sh and
exclude that toolset using the '-e' switch.

The toolset selected will not affect your benchmark scores.

linux-suse10-amd64            For 64-bit AMD64/EM64T Linux systems running
                              SuSE Linux 10 or later.
                              Built on SuSE Linux 10 with
                              GCC v4.1.0 (SUSE Linux)



=======================================================================
Attempting to install the linux-suse10-amd64 toolset...


Unpacking CPU2006 base files (129.8 MB)
```

```
Unpacking 400.perlbench benchmark and data files (61.5 MB)
Unpacking 401.bzip2 benchmark and data files (110.7 MB)
Unpacking 403.gcc benchmark and data files (43 MB)
Unpacking 410.bwaves benchmark and data files (0.1 MB)
Unpacking 416.gamess benchmark and data files (16.8 MB)
Unpacking 429.mcf benchmark and data files (6.9 MB)
Unpacking 433.milc benchmark and data files (0.6 MB)
Unpacking 434.zeusmp benchmark and data files (1.1 MB)
Unpacking 435.gromacs benchmark and data files (13 MB)
Unpacking 436.cactusADM benchmark and data files (3.3 MB)
Unpacking 437.leslie3d benchmark and data files (0.3 MB)
Unpacking 444.namd benchmark and data files (7.6 MB)
Unpacking 445.gobmk benchmark and data files (9.2 MB)
Unpacking 447.dealII benchmark and data files (70.2 MB)
Unpacking 450.soplex benchmark and data files (321 MB)
Unpacking 453.povray benchmark and data files (10.3 MB)
Unpacking 454.calculix benchmark and data files (26 MB)
Unpacking 456.hmmer benchmark and data files (57 MB)
Unpacking 458.sjeng benchmark and data files (0.4 MB)
Unpacking 459.GemsFDTD benchmark and data files (2.6 MB)
Unpacking 462.libquantum benchmark and data files (0.2 MB)
Unpacking 464.h264ref benchmark and data files (52.9 MB)
Unpacking 465.tonto benchmark and data files (6.9 MB)
Unpacking 470.lbm benchmark and data files (4.7 MB)
Unpacking 471.omnetpp benchmark and data files (2.8 MB)
Unpacking 473.astar benchmark and data files (5.8 MB)
Unpacking 481.wrf benchmark and data files (67.1 MB)
Unpacking 482.sphinx3 benchmark and data files (51.7 MB)
Unpacking 483.xalancbmk benchmark and data files (213.7 MB)
Unpacking 998.specrand benchmark and data files (4.1 MB)
Unpacking 999.specrand benchmark and data files (4.1 MB)

Checking the integrity of your source tree...


Checksums are all okay.

Unpacking binary tools for linux-suse10-amd64...

Checking the integrity of your binary tools...

Checksums are all okay.

Testing the tools installation (this may take a minute)

.....................................................................o.....................................
.....................................................................................................

Installation successful.  Source the shrc or cshrc in
/cpu2006
to set up your environment for the benchmark.
```

Step 6: Now, we change the current working directory from the install media to the location of the new SPEC CPU2006 tree. Since this user has a a Bourne compatible shell, `shrc` is sourced (for `csh` compatible shells, use `cshrc`).

Next, the config file `Example-linux64-amd64-gcc43+.cfg` has been picked as a starting point for this system. The gcc compiler might not provide the best possible score for this particular SUT (System Under Test), a laptop with an Intel Core i5 M 520 chip, but this config file provides a reasonable start in order to demonstrate that the SPEC tree is functional.

(Note that the term "amd64" in the config file name does not designate a chip from a particular manufacturer; rather, it designates an instruction set, variously known as "amd64", "EM64T", and "x86_64". The config file is an OK starting point for this SUT.)

```
$ cd /cpu2006/
$ . ./shrc
$ cd config
$ cp Example-linux64-amd64-gcc43+.cfg mytest.cfg
$ runspec --config=mytest.cfg --action=build --tune=base bzip2
runspec v6545 - Copyright 1999-2011 Standard Performance Evaluation Corporation
Using 'linux-suse10-amd64' tools
Reading MANIFEST... 22464 files
Loading runspec modules.................
Locating benchmarks...found 31 benchmarks in 6 benchsets.
Reading config file '/cpu2006/config/mytest.cfg'
```

```
Running "specperl /cpu2006/Docs/sysinfo" to gather system information.
Loading "http://www.spec.org/auto/cpu2006/current_version" for version check: OK

Retrieving flags file (/cpu2006/config/flags/Example-gcc4x-flags-revA.xml)...
Retrieving flags file (/cpu2006/config/flags/Example-linux-platform-revA.xml)...
Benchmarks selected: 401.bzip2
Compiling Binaries
  Building 401.bzip2 base gcc43-64bit default: (build_base_gcc43-64bit.0000)
specmake clean 2> make.clean.err | tee make.clean.out
rm -rf *.o  byoudoin.jpg.out input.combined.out input.program.out
find . \( -name \*.o -o -name '*.fppized.f*' -o -name '*.i' -o -name '*.mod' \) -print | xargs rm -rf
rm -rf bzip2
rm -rf bzip2.exe
rm -rf core
rm -rf
specmake build 2> make.err | tee make.out
/usr/bin/gcc -c -o spec.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   spec.c
/usr/bin/gcc -c -o blocksort.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   blocksort.c
/usr/bin/gcc -c -o bzip2.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   bzip2.c
bzip2.c: In function 'compressStream':
bzip2.c:484: warning: cast from pointer to integer of different size
bzip2.c:487: warning: assignment makes integer from pointer without a cast
bzip2.c: In function 'uncompressStream':
bzip2.c:611: warning: cast from pointer to integer of different size
bzip2.c:614: warning: assignment makes integer from pointer without a cast
/usr/bin/gcc -c -o bzlib.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   bzlib.c
bzlib.c: In function 'BZ2_bzWriteOpen':
bzlib.c:996: warning: cast from pointer to integer of different size
bzlib.c: In function 'BZ2_bzReadOpen':
bzlib.c:1172: warning: cast from pointer to integer of different size
bzlib.c: In function 'bzopen_or_bzdopen':
bzlib.c:1512: warning: cast from pointer to integer of different size
bzlib.c:1521: warning: cast from pointer to integer of different size
/usr/bin/gcc -c -o compress.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   compress.c
/usr/bin/gcc -c -o crctable.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   crctable.c
/usr/bin/gcc -c -o decompress.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   decompress.c
/usr/bin/gcc -c -o huffman.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   huffman.c
/usr/bin/gcc -c -o randtable.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   randtable.c
/usr/bin/gcc   -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64  spec.o blocksort.o bzip2.o bzlib.o compress.o
    crctable.o decompress.o huffman.o randtable.o                    -o bzip2
specmake options 2> options.err | tee options.out
COMP: /usr/bin/gcc -c -o options.o -DSPEC_CPU -DNDEBUG -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64   <source>
C: CC="/usr/bin/gcc"
C: COBJOPT="-c -o options"
P: CPUFLAGS="-DSPEC_CPU -DNDEBUG"
P: BENCH_FLAGS=""
P: BENCH_CFLAGS=""
O: OPTIMIZE=""
O: COPTIMIZE="-O2 -fno-strict-aliasing"
P: PORTABILITY="-DSPEC_CPU_LP64"
P: CPORTABILITY=""
O: EXTRA_CFLAGS=""
O: EXTRA_OPTIMIZE=""
O: EXTRA_COPTIMIZE=""
P: EXTRA_PORTABILITY=""
P: EXTRA_CPORTABILITY=""
LINK: /usr/bin/gcc   -O2 -fno-strict-aliasing  -DSPEC_CPU_LP64  <objects>          -o options
C: LD="/usr/bin/gcc"
O: COPTIMIZE="-O2 -fno-strict-aliasing"
P: PORTABILITY="-DSPEC_CPU_LP64"
C: LDOUT="-o options"

Build successes: 401.bzip2(base)          <<-- what we want to see

Build Complete

The log for this run is in /cpu2006/result/CPU2006.001.log

runspec finished at Wed Jul  6 22:30:27 2011; 12 total seconds elapsed
```

Just above, various compile and link commands may or may not be echoed to your screen, depending on the settings in your config file. At this point, we've accomplished a lot. The SPEC tree is installed, and we have verified that a benchmark can be compiled using the C compiler. (The sharp-eyed reader may notice some warnings above about casts of pointers. These warnings from the compiler have been reviewed by SPEC's project leader for 401.bzip2, who has determined that they will not affect operation of the benchmark.)

Step 8: Now try running a benchmark, using the minimal test workload. The test workload runs in a tiny amount of time and does a minimal verification that the benchmark executable can at least start up:

```
$ runspec --config=mytest.cfg --size=test --noreportable --tune=base --iterations=1 bzip2
runspec v6545 - Copyright 1999-2011 Standard Performance Evaluation Corporation
Using 'linux-suse10-amd64' tools
Reading MANIFEST... 22464 files
Loading runspec modules.................
Locating benchmarks...found 31 benchmarks in 6 benchsets.
Reading config file '/cpu2006/config/mytest.cfg'
Running "specperl /cpu2006/Docs/sysinfo" to gather system information.
Retrieving flags file (/cpu2006/config/flags/Example-gcc4x-flags-revA.xml)...
Retrieving flags file (/cpu2006/config/flags/Example-linux-platform-revA.xml)...
Benchmarks selected: 401.bzip2
Compiling Binaries
  Up to date 401.bzip2 base gcc43-64bit default


Setting Up Run Directories
  Setting up 401.bzip2 test base gcc43-64bit default: created (run_base_test_gcc43-64bit.0000)
Running Benchmarks
  Running 401.bzip2 test base gcc43-64bit default
/cpu2006/bin/specinvoke -d /cpu2006/benchspec/CPU2006/401.bzip2/run/run_base_test_gcc43-64bit.0000
    -e speccmds.err -o speccmds.stdout -f speccmds.cmd -C -q
/cpu2006/bin/specinvoke -E -d /cpu2006/benchspec/CPU2006/401.bzip2/run/run_base_test_gcc43-64bit.0000 -c 1
    -e compare.err -o compare.stdout -f compare.cmd
Success: 1x401.bzip2                          <<-- what we want to see
Producing Raw Reports
mach: default
  ext: gcc43-64bit
    size: test
      set: int
        format: raw -> /cpu2006/result/CINT2006.002.test.rsf
Parsing flags for 401.bzip2 base: done
Doing flag reduction: done
        format: flags -> /cpu2006/result/CINT2006.002.test.flags.html
        format: ASCII -> /cpu2006/result/CINT2006.002.test.txt
        format: CSV -> /cpu2006/result/CINT2006.002.test.csv
        format: HTML -> /cpu2006/result/CINT2006.002.test.html, /cpu2006/result/CINT2006.002.test.gif
      set: fp

The log for this run is in /cpu2006/result/CPU2006.002.log

runspec finished at Wed Jul  6 22:35:27 2011; 21 total seconds elapsed
```

Notice about 20 lines up the notation "`Success: 1x401.bzip2`". That is what we want to see.

Step 9: let's try running bzip2 with the real workload. This will take a while, over 11 minutes on the tested laptop running Linux.

```
$ runspec --config=mytest.cfg --size=ref --noreportable --tune=base --iterations=1 bzip2
runspec v6545 - Copyright 1999-2011 Standard Performance Evaluation Corporation
Using 'linux-suse10-amd64' tools
Reading MANIFEST... 22464 files
Loading runspec modules.................
Locating benchmarks...found 31 benchmarks in 6 benchsets.
Reading config file '/cpu2006/config/mytest.cfg'
Running "specperl /cpu2006/Docs/sysinfo" to gather system information.
Retrieving flags file (/cpu2006/config/flags/Example-gcc4x-flags-revA.xml)...
Retrieving flags file (/cpu2006/config/flags/Example-linux-platform-revA.xml)...
Benchmarks selected: 401.bzip2
Compiling Binaries
  Up to date 401.bzip2 base gcc43-64bit default


Setting Up Run Directories
  Setting up 401.bzip2 ref base gcc43-64bit default: created (run_base_ref_gcc43-64bit.0000)
Running Benchmarks
  Running 401.bzip2 ref base gcc43-64bit default
/cpu2006/bin/specinvoke -d
/cpu2006/benchspec/CPU2006/401.bzip2/run/run_base_ref_gcc43-64bit.0000 -e speccmds.err
```

```
   -o speccmds.stdout -f speccmds.cmd -C -q
/cpu2006/bin/specinvoke -E -d
/cpu2006/benchspec/CPU2006/401.bzip2/run/run_base_ref_gcc43-64bit.0000 -c 1 -e compare.err
   -o compare.stdout -f compare.cmd
Success: 1x401.bzip2                          <<-- what we want to see
Producing Raw Reports
mach: default
  ext: gcc43-64bit
    size: ref
      set: int
        format: raw -> /cpu2006/result/CINT2006.003.ref.rsf
Parsing flags for 401.bzip2 base: done
Doing flag reduction: done
        format: flags -> /cpu2006/result/CINT2006.003.ref.flags.html
        format: ASCII -> /cpu2006/result/CINT2006.003.ref.txt
        format: CSV -> /cpu2006/result/CINT2006.003.ref.csv
        format: HTML -> /cpu2006/result/CINT2006.003.ref.html, /cpu2006/result/CINT2006.003.ref.gif
      set: fp

The log for this run is in /cpu2006/result/CPU2006.003.log

runspec finished at Thu Jul  7 09:44:53 2011; 684 total seconds elapsed
$
```

Success with the real workload! So now let's look in the result directory and see what we find:

```
$ cd result
$ ls
CINT2006.002.test.csv         CINT2006.003.ref.csv         CPU2006.001.log
CINT2006.002.test.flags.html  CINT2006.003.ref.flags.html  CPU2006.002.log
CINT2006.002.test.gif         CINT2006.003.ref.gif         CPU2006.003.log
CINT2006.002.test.html        CINT2006.003.ref.html        invalid.gif
CINT2006.002.test.rsf         CINT2006.003.ref.rsf         lock.CPU2006
CINT2006.002.test.txt         CINT2006.003.ref.txt
$ grep runspec: *log
CPU2006.001.log:runspec: runspec --config=mytest.cfg --action=build --tune=base bzip2
CPU2006.002.log:runspec: runspec --config=mytest.cfg --size=test --noreportable --tune=base --iterations=1 bzip2
CPU2006.003.log:runspec: runspec --config=mytest.cfg --size=ref --noreportable --tune=base --iterations=1 bzip2
$
```

Notice the three separate sets of files: .001, .002, and .003

`CPU2006.001.log` has the log from the compile.

`CPU2006.002.log` has the log from running `401.bzip2` with the "test" input. The various outputs (`.csv`, `.html`, `.rsf`, `.txt`), and a compiler flags report (`.flags.html`), are all preceded by "`CINT2006`", because `401.bzip2` is one of the integer benchmarks. The tools also distinguish whether the input was a "test" input by putting that in the file name as well.

`CPU2006.003.log` has the log from running `401.bzip2` with the "ref" input. Once again, the various outputs and the configuration file (`.cfg`) all start with CINT2006.

Here is the complete `.txt` report from running 401.bzip2 ref:

```
$ cat CINT2006.003.ref.txt
################################################################################
#   INVALID RUN -- INVALID RUN -- INVALID RUN -- INVALID RUN -- INVALID RUN  #
#                                                                            #
# 'reportable' flag not set during run                                       #
# 471.omnetpp (base) did not have enough runs!                               #
# 456.hmmer (base) did not have enough runs!                                 #
# 445.gobmk (base) did not have enough runs!                                 #
# 458.sjeng (base) did not have enough runs!                                 #
# 429.mcf (base) did not have enough runs!                                   #
# 473.astar (base) did not have enough runs!                                 #
# 483.xalancbmk (base) did not have enough runs!                             #
# 400.perlbench (base) did not have enough runs!                             #
# 464.h264ref (base) did not have enough runs!                               #
# 462.libquantum (base) did not have enough runs!                            #
# 401.bzip2 (base) did not have enough runs!                                 #
# 403.gcc (base) did not have enough runs!                                   #
```

```
#                                                                               #
#   INVALID RUN -- INVALID RUN -- INVALID RUN -- INVALID RUN -- INVALID RUN  #
#################################################################################
                        SPEC(R) CINT2006 Summary
                      System Vendor System Model Name
          Test Sponsor: Test Sponsor (Optional, defaults to hw_vendor)
                        Thu Jul  7 09:33:29 2011

CPU2006 License: 0                                      Test date: Jul-2011
Test sponsor: Test Sponsor (Optional, defaults to hw_vendor)  Hardware availability: Dec-9999
Tested by:    (Optional, defaults to hw_vendor)  Software availability: May-2010

                                Estimated                       Estimated
                Base     Base       Base        Peak     Peak       Peak
Benchmarks      Ref.   Run Time     Ratio       Ref.   Run Time     Ratio
-------------- ------  ---------  ---------    ------  ---------  ---------
400.perlbench                          NR
401.bzip2         9650        666       14.5 *
403.gcc                                NR
429.mcf                                NR
445.gobmk                              NR
456.hmmer                              NR
458.sjeng                              NR
462.libquantum                         NR
464.h264ref                            NR
471.omnetpp                            NR
473.astar                              NR
483.xalancbmk                          NR
================================================================================
400.perlbench                          NR
401.bzip2         9650        666       14.5 *
403.gcc                                NR
429.mcf                                NR
445.gobmk                              NR
456.hmmer                              NR
458.sjeng                              NR
462.libquantum                         NR
464.h264ref                            NR
471.omnetpp                            NR
473.astar                              NR
483.xalancbmk                          NR
 Est. SPECint(R)_base2006                --
 Est. SPECint2006                                                    Not Run


                                  HARDWARE
                                  --------
            CPU Name: Intel Core i5 M 520
 CPU Characteristics:
             CPU MHz: 9999
                 FPU: Integrated
     CPU(s) enabled: 2 cores, 1 chip, 2 cores/chip, 2 threads/core
   CPU(s) orderable: 1,2 chips
      Primary Cache: 9999 KB I + 9999 KB D on chip per core
    Secondary Cache: 9999 KB I+D on chip per core
            L3 Cache: 9999 MB I+D on chip per chip
        Other Cache: None
             Memory: 3.666 GB fixme: If using DDR3, format is:
                     'N GB (M x N GB nRxn PCn-nnnnnR-n, ECC)'
     Disk Subsystem: 77 GB  add more disk info here
     Other Hardware: None


                                  SOFTWARE
                                  --------
   Operating System: Oracle Linux Server release 6.0
                     2.6.32-100.28.5.el6.x86_64
            Compiler: gcc, g++ & gfortran 4.3.4
       Auto Parallel: No
         File System: ext4
        System State: Run level 5 (add definition here)
       Base Pointers: 64-bit
       Peak Pointers: 64-bit
     Other Software: None


                                Submit Notes
                                ------------
   'numactl' was used to bind copies to the cores.
   See the configuration file for details.
```

```
                            Operating System Notes
                            ----------------------
     'ulimit -s unlimited' was used to set environment stack size

                               Platform Notes
                               --------------
      Sysinfo program (/cpu2006/Docs/sysinfo)
      on jhenning-t410-oel Thu Jul  7 09:33:31 2011

      This section contains SUT (System Under Test) info as seen by
      some common utilities.  To remove or add to this section, see:
        http://www.spec.org/cpu2006/Docs/config.html#sysinfo

      From /proc/cpuinfo
         model name : Intel(R) Core(TM) i5 CPU M 520 @ 2.40GHz
            1 chips
            physical 0: cores 0 2
            4 threads
            cache size : 3072 KB

      From /proc/meminfo
         MemTotal:        3844200 kB
         HugePages_Total:       0
         Hugepagesize:       2048 kB

      /usr/bin/lsb_release -d
         Oracle Linux Server release 6.0

      From /etc/*release* /etc/*version*
         oracle-release: Oracle Linux Server release 6.0
         redhat-release: Red Hat Enterprise Linux Server release 6.0 (Santiago)
         system-release: Oracle Linux Server release 6.0
         system-release-cpe: cpe:/o:oracle:oracle_linux:6server:ga:server

      uname -a:
         Linux jhenning-t410-oel 2.6.32-100.28.5.el6.x86_64 #1 SMP Wed Feb 2 18:40:23
         EST 2011 x86_64 x86_64 x86_64 GNU/Linux

      run-level 5 Jul 6 20:30

      SPEC is set to: /home/jhenning/cpu2006/kit117
         Filesystem     Type    Size  Used Avail Use% Mounted on
         /dev/mapper/vg_oel6-lv_root
                        ext4     77G  7.2G   66G  10% /

      (End of data from sysinfo program)

                            Base Compiler Invocation
                            ------------------------
C benchmarks:

 401.bzip2: /usr/bin/gcc


                             Base Portability Flags
                             ----------------------
 401.bzip2: -DSPEC_CPU_LP64


                            Base Optimization Flags
                            -----------------------
C benchmarks:

 401.bzip2: -O2 -fno-strict-aliasing


                               Base Other Flags
                               ----------------
C benchmarks:

 401.bzip2: No flags used


    SPEC and SPECint are registered trademarks of the Standard Performance
    Evaluation Corporation.  All other brand and product names appearing
    in this result are trademarks or registered trademarks of their
    respective holders.
##############################################################################
#   INVALID RUN -- INVALID RUN -- INVALID RUN -- INVALID RUN -- INVALID RUN  #
```

```
#                                                                              #
# 'reportable' flag not set during run                                         #
# 471.omnetpp (base) did not have enough runs!                                 #
# 456.hmmer (base) did not have enough runs!                                   #
# 445.gobmk (base) did not have enough runs!                                   #
# 458.sjeng (base) did not have enough runs!                                   #
# 429.mcf (base) did not have enough runs!                                     #
# 473.astar (base) did not have enough runs!                                   #
# 483.xalancbmk (base) did not have enough runs!                               #
# 400.perlbench (base) did not have enough runs!                               #
# 464.h264ref (base) did not have enough runs!                                 #
# 462.libquantum (base) did not have enough runs!                              #
# 401.bzip2 (base) did not have enough runs!                                   #
# 403.gcc (base) did not have enough runs!                                     #
#                                                                              #
#   INVALID RUN -- INVALID RUN -- INVALID RUN -- INVALID RUN -- INVALID RUN    #
################################################################################
--------------------------------------------------------------------------------
For questions about this result, please contact the tester.
For other inquiries, please contact webmaster@spec.org.
Copyright 2006-2011 Standard Performance Evaluation Corporation
Tested with SPEC CPU2006 v117.
Report generated on Thu Jul  7 09:44:53 2011 by CPU2006 ASCII formatter v6400.
```

Done. The suite is installed, and we can run at least one benchmark for real (see the report of the time spent in 401.bzip2 above).

Appendix 1: the DVD drive is on system A, but I want to install on system B. What do I do?

If the title of this section describes your situation, you basically have two choices.

1. Network mount: You can mount the device over the network and do the installation remotely.
2. Tar file: You can install from the tar file

1. Network mount

You might be able to mount the DVD on one system and use network services to make it available on other systems.

Please note that the SPEC CPU2006 license agreement does not allow you to post the DVD on any public server. If your institution has a SPEC CPU2006 license, then it's fine to post it on an internal server that is accessible only to members of your institution.

Whether you attempt a network mount will probably depend on:

- available network hardware,
- the networking services provided by your operating system, and
- the security policies of your IT department (if any).

If your network environment allows easy cross-system mounting, or if you feel brave about reading manpages, you can use a network mount for the installation. Otherwise, you can fall back on the tar file.

Network mount, easy:
for example, System A Solaris/Opteron + System B Solaris/SPARC

Your operating system may be configured to automatically mount the drive and automatically make it visible to other network systems, or may make it visible with minimal user intervention. During one set of testing, system A (with the DVD drive) was an Opteron-based system running Solaris 10. The SPEC CPU2006 DVD was inserted. The operating system mounted it automatically, and from a terminal window, a (non-privileged) user entered the Solaris

```
share
```

command to make it visible to other hosts.

On System B, a Solaris SPARC system, a non-privileged user typed:

```
cd /net/systemA/cdrom/spec_cpu2006
./install.sh
```

and the installation proceeded normally, picking up from step 5, above.

Network mount, medium difficulty:
for example, System A Solaris/Opteron + System B Tru64 Unix/Alpha

Subsequent to the tests of the previous paragraphs, the DVD drive on System A (Solaris/Opteron) was also visible to a system running Compaq Tru64 UNIX V5.1A. But in this case, a little assistance was needed from the privileged (root) account on system B:

```
echo "systemA.domain.com:/cdrom/spec_cpu2006 /systemA nfs ro,bg,soft 0 0" >> /etc/fstab
mkdir /systemA
/usr/sbin/mount /systemA
```

Then, the non-privileged user was able to say:

```
cd /systemA
./install.sh
```

and once again the installation proceeded normally, picking up from step 5, above.

Network mount, a bit harder:
for example, System A SuSE/x86 + System B Mac OS X/PowerPC

The SPEC CPU2006 DVD was also inserted into a system running SuSE Linux 9.0, and used from a Mac OS X PowerBook. On both these systems, there are probably automatic tools that would have accomplished the following more quickly, but the tester happened to read the manpages in the particular order that he happened to read them in. The following succeeded:

On System A, root added

```
/dev/cdrom  /cd  iso9660  ro,user,noauto,unhide
```

to `/etc/fstab` as suggested by `man mount`; the DVD was inserted; and the user typed `mount /cd`. On System A, root also added:

```
/cd  192.168.0.0/24(ro,insecure,no_root_squash,sync)
```

to `/etc/exports`, and then typed:

```
exportfs -r
rpc.nfsd -p 8
rpc.mountd
cat /var/lib/nfs/etab
```

On System B, root typed:

```
mkdir /remote
mount -t nfs 192.168.0.106:/cd /remote
```

Finally, the user typed

```
cd /remote
./install.sh
```

and installation continued as normal, with step 5.

2. Tar file

If the DVD drive is on a system other than the one where you wish to do the installation, and if you do not wish to try to get a network mount working, then the final fallback is to use the compressed tarfile. If you choose this option, please carefully observe the warnings.

a. Go to the system with the DVD drive ("System A"). Insert the SPEC CPU2006 DVD, and, if required, issue a mount command.

b. From a terminal window (aka command window), `cd` to the top level directory on the DVD.

c. You are going to retrieve five things from the DVD. First, find the large tarfile and its corresponding md5 file:

```
cd install_archives
ls -l cpu2006.tar.xz*
```

or, if System A is a Windows system, then:

```
cd install_archives
dir cpu2006.tar.xz*
```

In either case, you should see one moderately large file > 500MB, `cpu2006.tar.xz`, and a small file associated with it that contains a checksum, `cpu2006.tar.xz.md5`.

> If you don't see the above files, try looking for `cpu*tar*`. The name might change if, for example, a maintenance update of SPEC CPU2006 changes the name slightly to indicate an updated version.

Do whatever is required in order to transfer both files intact to the system where you wish to do the installation ("System B"). If you use `ftp`, do not forget to use image (binary) mode. For example:

```
$ ftp
ftp> op systemB
Name: imauser
Password:
ftp> cd /kits
ftp> bin    <-------- important
200 Type set to I.
ftp> put cpu2006.tar.xz
ftp> put cpu2006.tar.xz.md5
```

Please note that the SPEC CPU2006 license agreement does not allow you to post the above file on any public ftp server. If your institution has a SPEC CPU2006 license, then it's fine to post it on an internal server that is accessible only to members of your institution.

d. Next, you are going to look on the DVD for versions of `specxz`, `specmd5sum`, and `spectar` that are compatible with system B. Please do not use the `tar` supplied by your operating system unless you are sure that it can handle long path names. Many commonly-supplied `tar` utilities cannnot.

Please do not use Windows Zip utilities, as these will not preserve line endings.

If you have GNU tar and the genuine xz, then you can use those; otherwise, please hunt around on the DVD to find prebuilt versions that are compatible with your environment, like so:

```
$ cd /media/SPEC_CPU2006/
$ cd tools/bin
$ ls
aix5L-ppc          linux-debian31-ppc   linux-suse10-ia32   solaris10-x86
aix5L-ppc64        linux-redhat72-ia32  linux-suse10-ppc64  solaris-sparc
hpux11iv31-parisc  linux-rhas4r4-ia64   macosx              solaris-x86
hpux11iv3-ipf      linux-sles9-ia64     macosx-ppc          tru64-alpha
irix6.5-mips       linux-suse10-amd64   solaris10-sparc     windows-i386
$ cd aix5L-ppc
$ cat description
For PowerPC systems running AIX 5L V5.3 or later
                            Built on AIX 5L 5300-02 with the
                            IBM XL C/C++ for AIX V9.0.0.25 compiler
$ ls -g spec*
-r-xr-xr-x. 1 imauser  47596 Apr 22 18:30 specmd5sum
-r-xr-xr-x. 1 imauser 537176 Apr 22 18:29 spectar
-r-xr-xr-x. 1 imauser 233777 Apr 22 18:29 specxz
$
```

Once you've found the right versions of `specxz`, `specmd5sum`, and `spectar` for the system where you intend to install (system B), transfer them to system B using the same methods that you used for the big tarfile.

e. On system B, use `specmd5sum` to check that the file transfer worked correctly. In this example, we assume that you have placed all 5 of the files mentioned above in the `/kits` directory:

```
$ cd /kits
$ chmod +x spec*
$ specmd5sum -c cpu2006.tar.xz.md5
cpu2006.tar.xz: OK
```

f. Unpack the tarfile, like so:

```
$ cd /mybigdisk
$ mkdir cpu2006
$ cd cpu2006
$ /kits/specxz -dc /kits/cpu2006.tar.xz | /kits/spectar -xf -
```

Be patient: it will take a bit of time to unpack! It might take 15 minutes, depending on the speed of your processor and disks. Go for a coffee break.

g. Now, at last, type `./install.sh` and pick up with step 5, above. Your output will be similar, but not identical, to the output shown in step 5 above: you won't see the "Unpacking xxxx" messages, because you already did the unpacking.

Note that the directory where you unpack the tarfile will be the directory you install FROM and also the directory you install TO. **This is normal, and expected, for a tarfile installation.**

You will see a question similar to this:

```
Installing FROM /mybigdisk/cpu2006
Installing TO /mybigdisk/cpu2006

Is this correct? (Please enter 'yes' or 'no')
yes
```

If you enter "no", installation will stop. If you try to install TO another directory, using the `-d` flag, the installation will **not** succeed when using the tar file method.

Appendix 2: Uninstalling SPEC CPU2006

At this time, SPEC does not provide an uninstall utility for SPEC CPU2006. Confusingly, there is a file named `uninstall.sh` in the top directory, but it does not remove the whole product; it only removes the SPEC tool set, and does not affect the benchmarks (which consume the bulk of the disk space).

To remove SPEC CPU2006 on Windows systems, select the top directory in Windows Explorer and delete it.

To remove SPEC CPU2006 on Unix systems, use `rm -Rf` on the directory where you installed the suite, for example:

```
rm -Rf /home/cs3000/saturos/spec/cpu2006
```

If you have been using the output_root feature, you will have to track those down separately. Therefore, prior to removing the tree,, you might want to look for mentions of output root, for example:

```
Windows:
    cd %SPEC%\config
    findstr output_root *cfg

Unix:
    cd $SPEC/config
    grep output_root *cfg
```

Note: instead of deleting the entire directory tree, some users find it useful to keep the `config` and `result` subdirectories, while deleting everything else.
