# A PostgreSQL development environment

Peter Eisentraut

peter@eisentraut.org
@petereisentraut

## The plan

- tooling
- building
- testing
- developing
- documenting
- maintaining

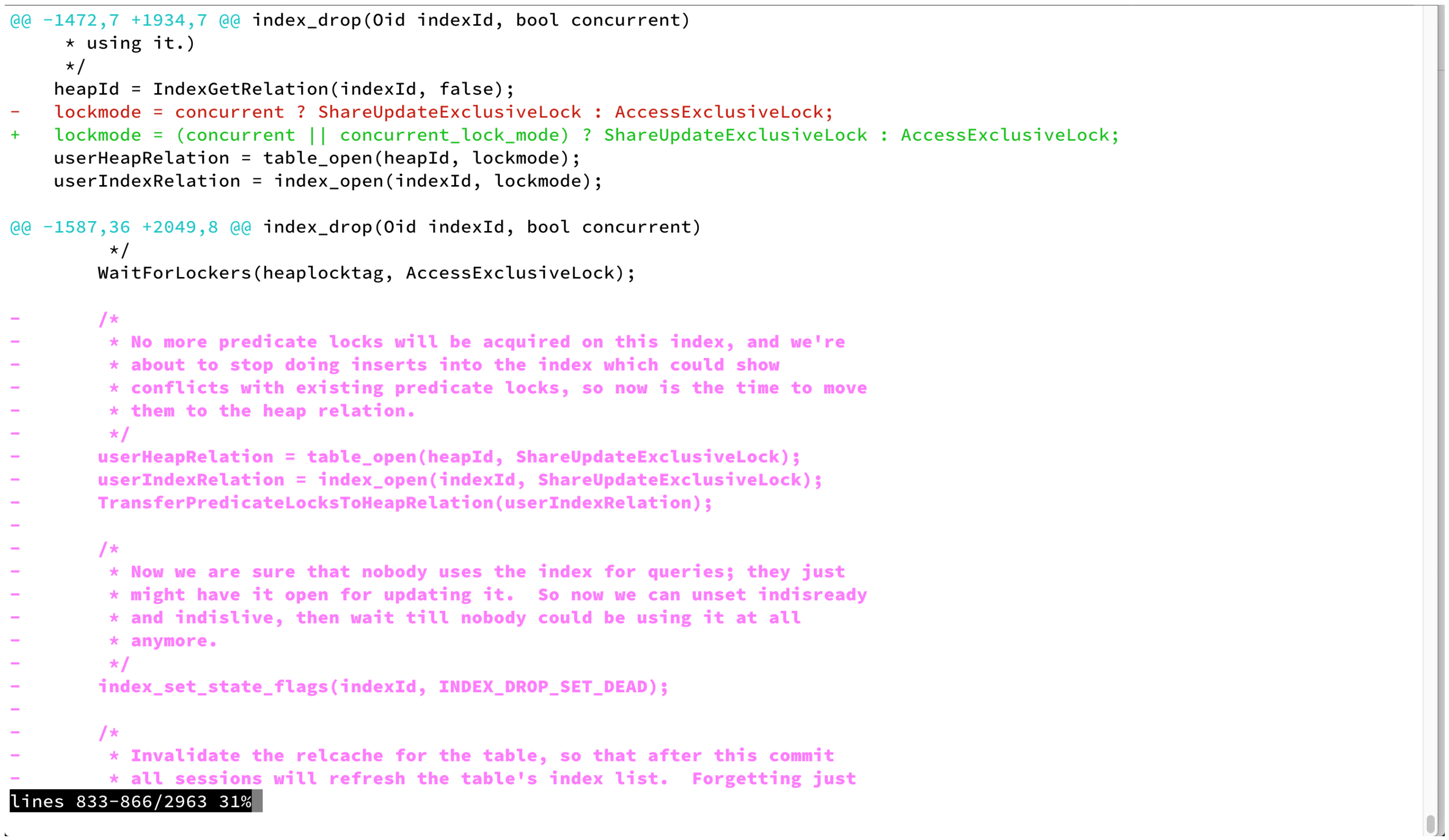
# Tooling: Git commands

Useful commands:

- `git add` (`-N`, `-p`)
- `git am`
- `git apply`
- `git bisect`
- `git blame -w`
- `git branch` (`--contains`, `-d`, `-D`, `--list`, `-m`, `-r`)
- `git checkout` (file, branch, `-b`)
- `git cherry-pick`
- `git clean` (`-f`, `-d`, `-x`, `-n`)
- `git commit` (`--amend`, `--reset`, `--fixup`, `-a`)
- `git diff`
- `git ls-files`
- `git format-patch`
- `git grep` (`-i`, `-w`, `-W`)
- `git log` (`-S`, `--grep`)
- `git merge`
- `git pull`
- `git push` (`-n`, `-f`, `-u`)
- `git rebase`
- `git reset` (`--hard`)
- `git show`
- `git status`
- `git stash`
- `git tag`

## Tooling: Git configuration

```
[diff]
        colorMoved = true
        colorMovedWS = allow-indentation-change
[log]
        date = local
[merge]
        conflictStyle = diff3
[rebase]
        autosquash = true
[stash]
        showPatch = true
[tag]
        sort = version:refname
[versionsort]
        suffix = "_BETA"
        suffix = "_RC"
```



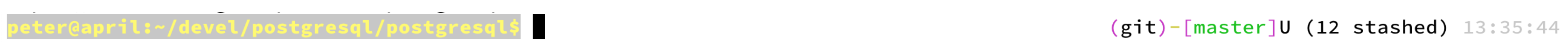
```
@@ -1472,7 +1934,7 @@ index_drop(Oid indexId, bool concurrent)
      * using it.)
      */
    heapId = IndexGetRelation(indexId, false);
-   lockmode = concurrent ? ShareUpdateExclusiveLock : AccessExclusiveLock;
+   lockmode = (concurrent || concurrent_lock_mode) ? ShareUpdateExclusiveLock : AccessExclusiveLock;
    userHeapRelation = table_open(heapId, lockmode);
    userIndexRelation = index_open(indexId, lockmode);

@@ -1587,36 +2049,8 @@ index_drop(Oid indexId, bool concurrent)
         */
        WaitForLockers(heaplocktag, AccessExclusiveLock);

-       /*
-        * No more predicate locks will be acquired on this index, and we're
-        * about to stop doing inserts into the index which could show
-        * conflicts with existing predicate locks, so now is the time to move
-        * them to the heap relation.
-        */
-       userHeapRelation = table_open(heapId, ShareUpdateExclusiveLock);
-       userIndexRelation = index_open(indexId, ShareUpdateExclusiveLock);
-       TransferPredicateLocksToHeapRelation(userIndexRelation);
-
-       /*
-        * Now we are sure that nobody uses the index for queries; they just
-        * might have it open for updating it.  So now we can unset indisready
-        * and indislive, then wait till nobody could be using it at all
-        * anymore.
-        */
-       index_set_state_flags(indexId, INDEX_DROP_SET_DEAD);
-
-       /*
-        * Invalidate the relcache for the table, so that after this commit
-        * all sessions will refresh the table's index list.  Forgetting just
lines 833-866/2963 31%
```

## Tooling: Git aliases

```
[alias]
check-whitespace = \
  !git diff-tree --check $(git hash-object -t tree /dev/null) HEAD ${1:-$GIT_PREFIX}
find = !git ls-files "*/$1"
gtags = !git ls-files | gtags -f -
st = status
tags = tag -l
wdiff = diff --color-words=[[:alnum:]]+
wshow = show --color-words
```

# Tooling: Git shell integration

zsh vcs_info



```
peter@april:~/devel/postgresql/postgresql$                    (git)-[master]U (12 stashed) 13:35:44
```

## Tooling: Editor

- Custom settings for PostgreSQL sources exist.
- Advanced options:
  - whitespace checking
  - automatic spell checking (flyspell)
  - automatic building (flycheck, flymake)
  - symbol lookup (”tags”)

# Tooling: Shell settings

https://superuser.com/questions/521657/zsh-automatically-set-environment-variables-for-a-directory

```
zstyle ':chpwd:profiles:/*/*/devel/postgresql(|/|/*)' profile postgresql

chpwd_profile_postgresql()
{
  chpwd_profile_default
  export EMAIL="peter@eisentraut.org"
  alias nano='nano -T4'
  LESS="$LESS -x4"
}
```

## Tooling — Summary

Your four friends:

- vcs
- editor
- shell
- terminal

## Building: How to run configure

```
./configure
-C/--config-cache
--prefix=$(cd .. && pwd)/pg-install
--enable-debug --enable-depend --enable-cassert
--enable-tap-tests
--with-pgport=65432
```

enable all the extra features (`--with-*`, `--enable-*`)

wrap all that into a shell script or alias (`pgconfigure`)

## Building: Always build with -Werror

```
if [ ! -e $srcdir/src/Makefile.custom ]; then
    cat <<EOF >$srcdir/src/Makefile.custom
COPT = -Werror
EOF
fi
```

(also put into `pgconfigure`)

## Building: ccache

Always install `ccache`.

## Building: make

```
make -j8 -k
make world -j8 -k
```

alternatively:

```
make -j -l8 -k
make world -j -l8 -k
```

(Find what works best for you.)

## Building: make install

parallel make is also applicable here

http://petereisentraut.blogspot.com/2012/03/postgresql-make-install-times.html

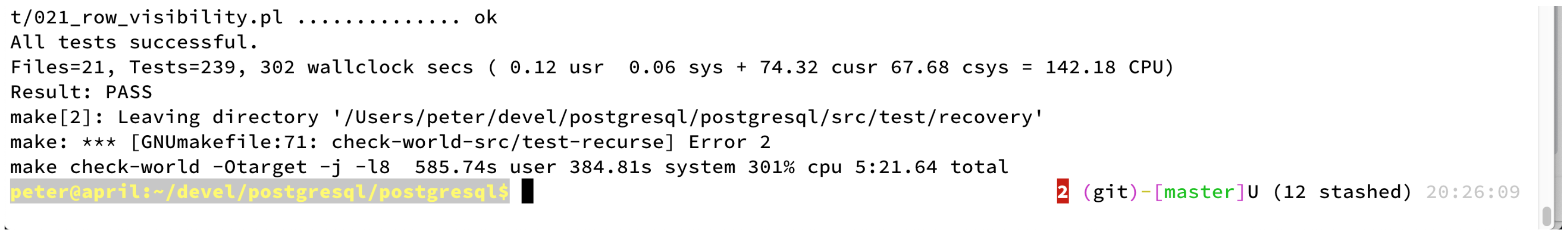
## Building — Summary

”pgconfigure” + parallel make + ccache

## Testing: make check

```
make check
make check-world -Otarget -j3
make check-world -Otarget -j -l8

make check NO_TEMP_INSTALL=1
```

## Testing: Shell exit status



```
t/021_row_visibility.pl .............. ok
All tests successful.
Files=21, Tests=239, 302 wallclock secs ( 0.12 usr  0.06 sys + 74.32 cusr 67.68 csys = 142.18 CPU)
Result: PASS
make[2]: Leaving directory '/Users/peter/devel/postgresql/postgresql/src/test/recovery'
make: *** [GNUmakefile:71: check-world-src/test-recurse] Error 2
make check-world -Otarget -j -l8  585.74s user 384.81s system 301% cpu 5:21.64 total
peter@april:~/devel/postgresql/postgresql$                                          2 (git)-[master]U (12 stashed) 20:26:09
```

## Testing: Extra tests

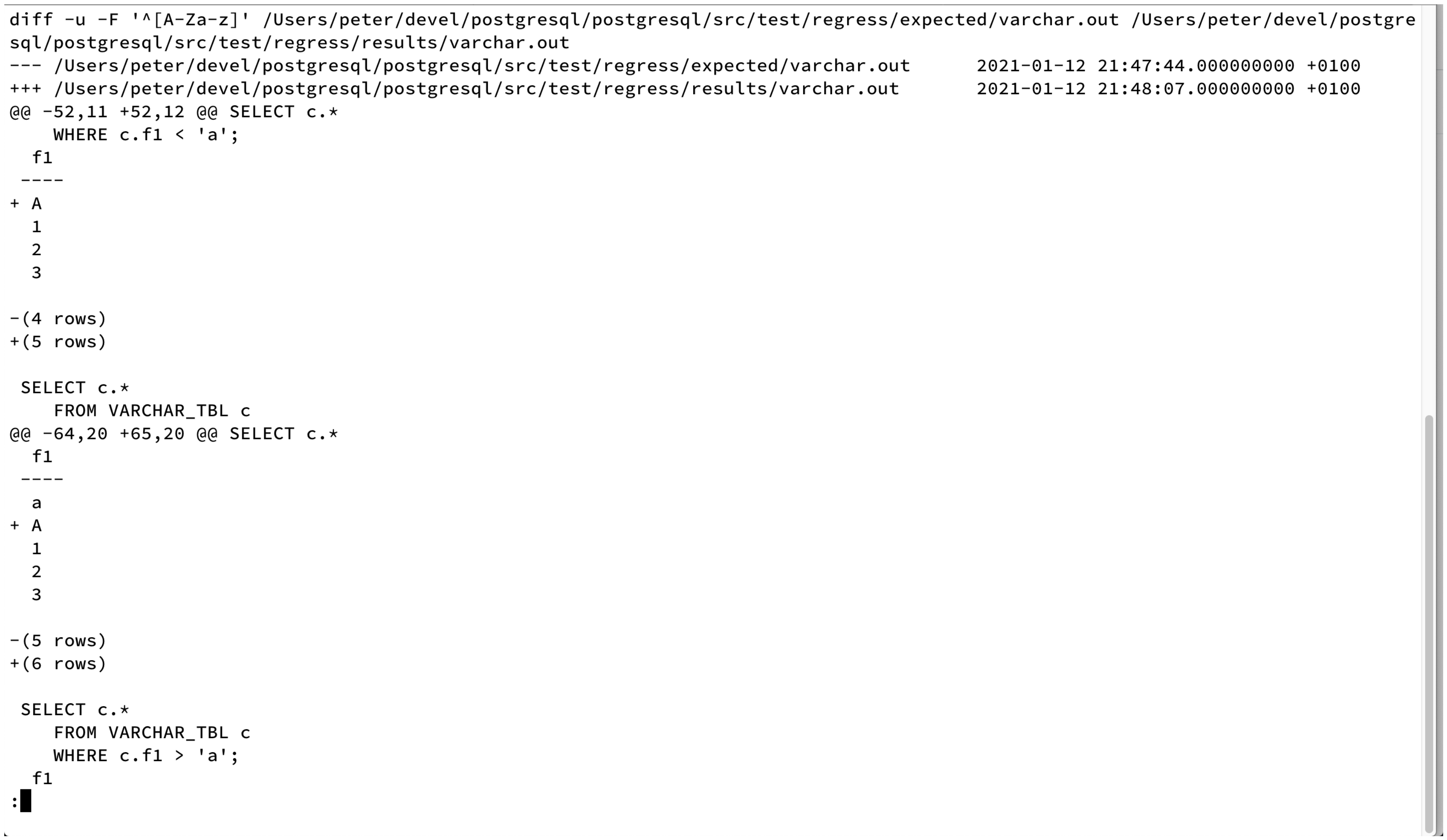
```
export PG_TEST_EXTRA='kerberos ldap ssl'
```

## Testing: Regression tests diff format

```
export PG_REGRESS_DIFF_OPTS="-u -F '^[A-Za-z]'"
```

# Testing: Regression tests diff in color

```
$ less src/test/regress/regression.diffs
```



```
diff -u -F '^[A-Za-z]' /Users/peter/devel/postgresql/postgresql/src/test/regress/expected/varchar.out /Users/peter/devel/postgre
sql/postgresql/src/test/regress/results/varchar.out
--- /Users/peter/devel/postgresql/postgresql/src/test/regress/expected/varchar.out      2021-01-12 21:47:44.000000000 +0100
+++ /Users/peter/devel/postgresql/postgresql/src/test/regress/results/varchar.out       2021-01-12 21:48:07.000000000 +0100
@@ -52,11 +52,12 @@ SELECT c.*
    WHERE c.f1 < 'a';
   f1
  ----
+ A
  1
  2
  3

-(4 rows)
+(5 rows)

 SELECT c.*
    FROM VARCHAR_TBL c
@@ -64,20 +65,20 @@ SELECT c.*
  f1
 ----
  a
+ A
  1
  2
  3

-(5 rows)
+(6 rows)

 SELECT c.*
    FROM VARCHAR_TBL c
    WHERE c.f1 > 'a';
  f1
:
```

## Testing: Regression tests diff in color

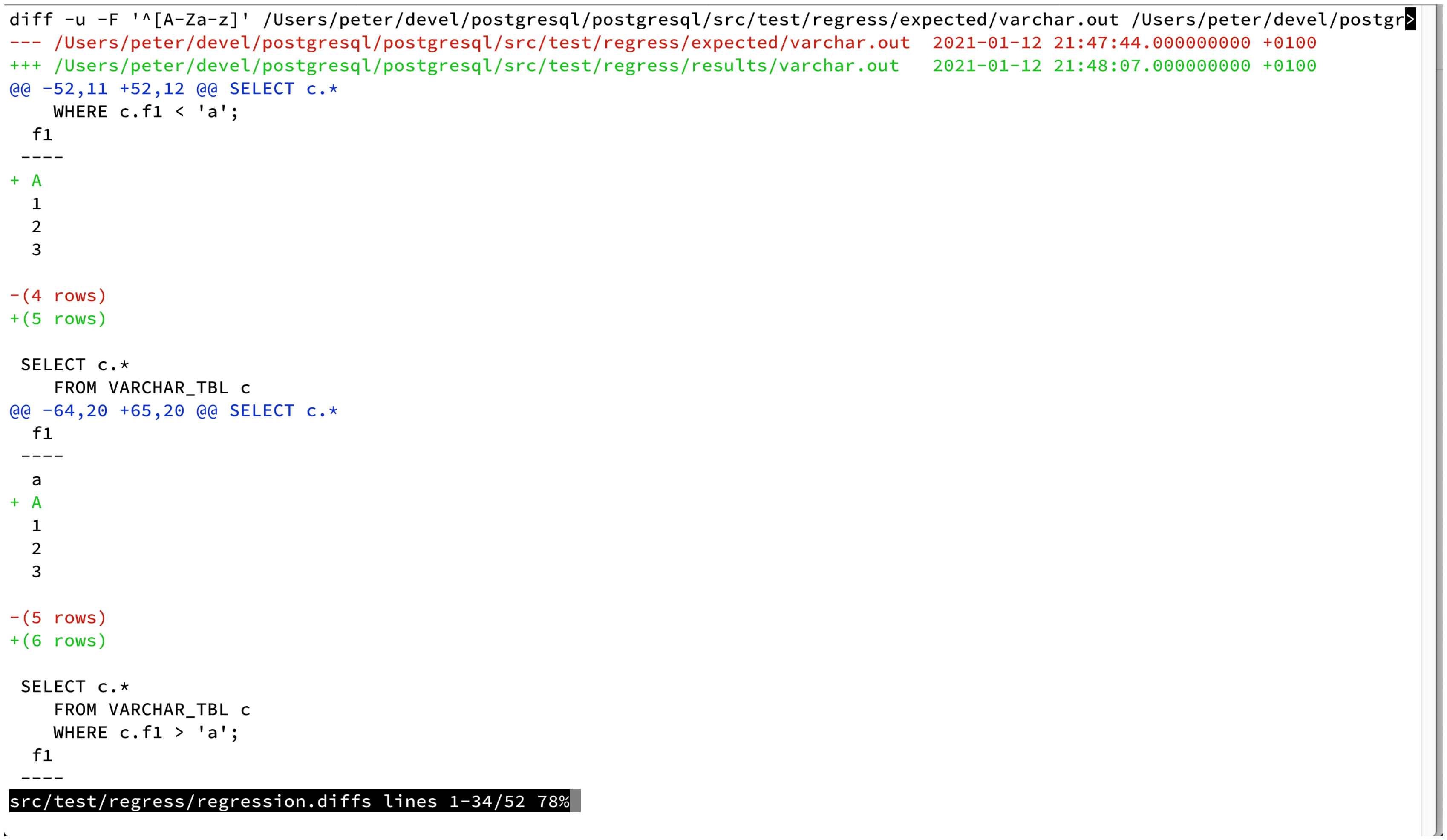
```
export LESS='-R'
export LESSOPEN="$HOME/.lessopen %s"
export LESSCLOSE="$HOME/.lessclose %s %s"

~/.lessopen

#!/bin/sh
infile=$1
tempfile=$(mktemp)
source-highlight -fesc -oSTDOUT -i "$infile" 2>/dev/null >$tempfile
echo "$tempfile"

~/.lessclose

#!/bin/sh
rm "$2"
```

## Testing: Regression tests diff in color

```
~/.lessopen

#!/bin/sh
infile=$1
tempfile=$(mktemp)
(
case $(basename "$infile") in
    *.diffs) opt='-s diff';;
    ...
    *.sgml) opt='-s xml';;
    *.xsl) opt='-s xml';;
esac
source-highlight -fesc -oSTDOUT -i "$infile" $opt 2>/dev/null
) >$tempfile
echo "$tempfile"
```

## Testing: Regression tests diff in color

```
$ less src/test/regress/regression.diffs
```



```
diff -u -F '^[A-Za-z]' /Users/peter/devel/postgresql/postgresql/src/test/regress/expected/varchar.out /Users/peter/devel/postgr>
--- /Users/peter/devel/postgresql/postgresql/src/test/regress/expected/varchar.out  2021-01-12 21:47:44.000000000 +0100
+++ /Users/peter/devel/postgresql/postgresql/src/test/regress/results/varchar.out   2021-01-12 21:48:07.000000000 +0100
@@ -52,11 +52,12 @@ SELECT c.*
    WHERE c.f1 < 'a';
   f1
  ----
+ A
  1
  2
  3

-(4 rows)
+(5 rows)

 SELECT c.*
    FROM VARCHAR_TBL c
@@ -64,20 +65,20 @@ SELECT c.*
   f1
  ----
  a
+ A
  1
  2
  3

-(5 rows)
+(6 rows)

 SELECT c.*
    FROM VARCHAR_TBL c
    WHERE c.f1 > 'a';
   f1
  ----
src/test/regress/regression.diffs lines 1-34/52 78%
```

## Testing: Simplify manual testing?

```
export PGDATABASE=postgres
```

## Testing: Core files

Linux:

```
kernel.core_pattern = core.%e.%p
```

or see `systemd-coredump(8)`

## Testing: Updating regression test files

Easy:

```
cp results/json.out expected/
```

Trickier:

```
merge expected/json_encoding_1.out expected/json_encoding.out \
  results/json_encoding.out
cp results/json_encoding.out expected/json_encoding.out

merge output/largeobject.source expected/largeobject.out \
  results/largeobject.out
```

https://wiki.postgresql.org/wiki/Regression_test_authoring#Updating_an_existing_regression_test

## Testing — Summary

- parallel make
- pg_regress tweaks
- could be easier

## Developing: Exploring

recommended: GNU Global

(ignore `src/tools/make_*tags`)

```
$ gtags
$ global elog
$ less -t elog
```

http://peter.eisentraut.org/blog/2016/07/20/using-gnu-global-with-postgresql/

## Developing: Test coverage

```
./configure ... --enable-coverage
make ...
make check ...
make coverage-html
$BROWSER coverage/index.html
```

https://coverage.postgresql.org/

## Developing: Emailing patches

```
git format-patch master
git format-patch -1

git format-patch master -v1, -v2, …
git format-patch master --base
master
```

https://www.2ndquadrant.com/en/blog/maintaining-feature-branches-submitting-patches-git/

## Developing: Useful tools

some useful things in `src/tools/`:

- pgindent
- perlcheck
- pginclude

## Documenting: Editor setup

- Configs for some editors are included.
- Syntax highlighting, tab completion, and well-formedness checking should work.

## Documenting: Faster builds

```
make html XSLTPROCFLAGS='--stringparam rootid pageinspect'
```

https://www.postgresql.org/message-id/1c91cb0c-0744-a1eb-d6ea-e952bc625fa8@2ndquadrant.com

## Maintaining: Multiple checkouts

I still use `git new-workdir`.
I can’t get `git worktree` to work well.

## Maintaining: Multiple branches

Suggestion: `myrepos` (`mr`)

```
~/.mrconfig:

[devel/postgresql/postgresql]
checkout = git clone git://git.postgresql.org/git/postgresql.git

[devel/postgresql/pg13/postgresql]
checkout = git new-workdir ../postgresql postgresql REL_13_STABLE

[devel/postgresql/pg12/postgresql]
checkout = git new-workdir ../postgresql postgresql REL_12_STABLE

...

~/devel/postgresql$ mr up
```

## Maintaining: Different ports

Idea:

```
./configure ... --with-pgport=$((50000 + $(pwd | cksum | cut -d ' ' -f 1) % 9999))

./configure ... --with-pgport=$((50000 + $(stat -c '%i' .) % 9999))
```

## Maintaining — Summary

Organize and automate all the checkouts.

## Conclusion

Keep the tools sharp!