Using Q editor macros to generate preprocessed Source Code without headers

Part 1: a basic implementation

Steps involved

- “Instrument” source code with comments showing where the headers are
- Build from instrumented source and keep the preprocessed files
- Edit header lines out of preprocessed files
- (on Day 3) Generate a shadow directory tree

Implementation (1/4)

- Create a new VCS branch to work in
- Change to the new branch
- Instrument the source (insert a ***cut here*** comment after the last #include in each file):

```
find . -name '*.c'|xargs q -oniu,cpp.qm^J^N2
```

Implementation (2/4)

- Set up special build flags (*temps/* is an example):

```
CFLAGS='-P -C -dumpdir temps/ -save-temps' ./configure
```

  -P suppresses line numbers; *-C* keeps comments (could use *-CC* to keep comments in macros)

- Clear the decks:

```
make -j$(($(nproc)+1)) clean;rm -rf temps; mkdir temps
```

Implementation (3/4)

- Do the build:

```
make -j$(($(nproc)+1))
```

- Remove unwanted assembler files:

```
rm temps/*.s
```

- You don’t want binaries either:

```
make -j$(($(nproc)+1)) clean
```

Implementation (4/4)

- Remove headers from .i files:

```
q -oiu,cpp.qm^J^N3 temps/*.i
```

If the last #include was #ifdef’d out, the ***cut here*** comment will be gone also. Need to edit out #include file contents manually later.

Compare files

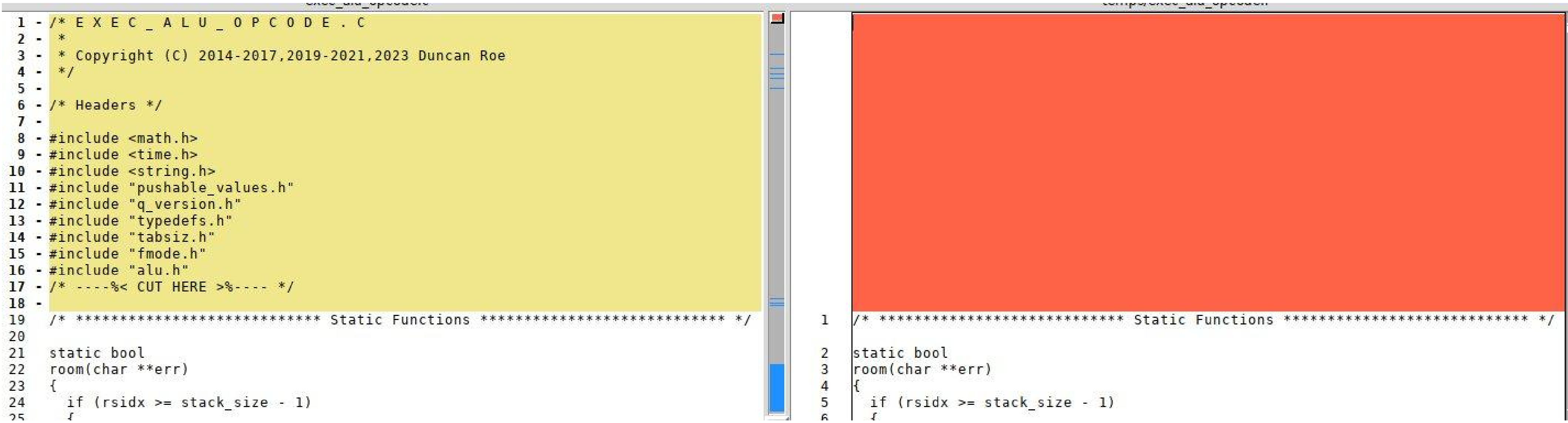
- Suggest using *tkdiff* to easily skip over uninteresting diffs
- cpp mangles white space a lot, so ignore whitespace and blank lines (e.g. **diff -Wb**)

Using Q editor macros to generate preprocessed Source Code without headers

Part 2: dealing with included .c files

Where we got to last time

Delete everything up to end of last #include



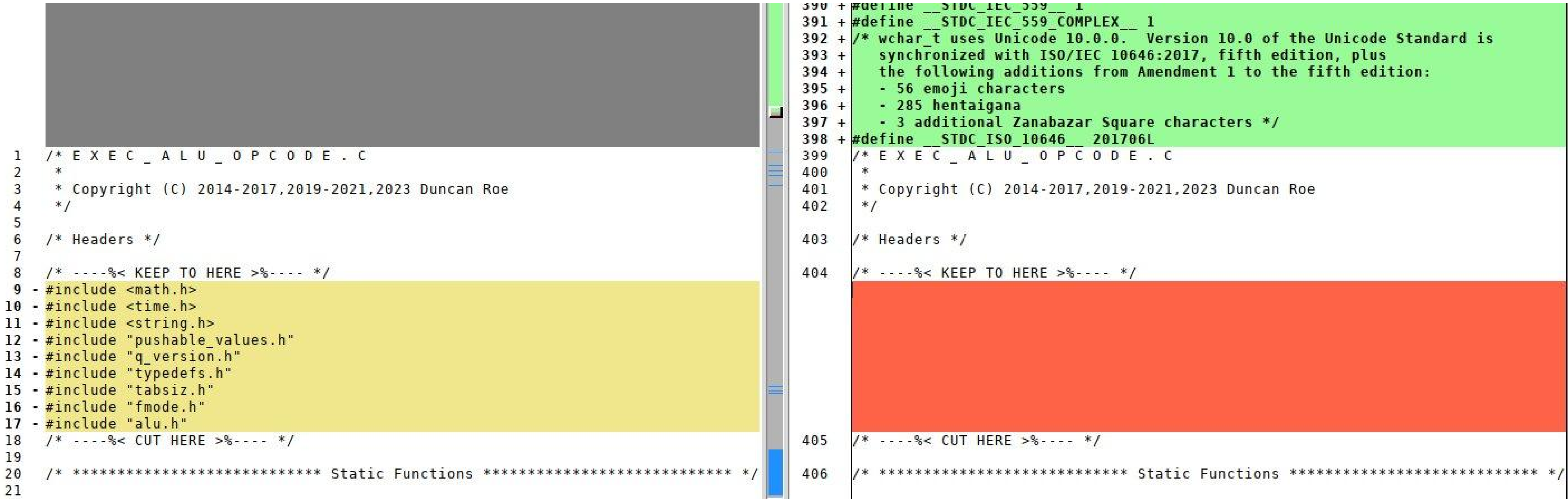
```
/* E X E C _ A L U _ O P C O D E . C
 *
 * Copyright (C) 2014-2017,2019-2021,2023 Duncan Roe
 */

/* Headers */

#include <math.h>
#include <time.h>
#include <string.h>
#include "pushable_values.h"
#include "q_version.h"
#include "typedefs.h"
#include "tabsiz.h"
#include "fmode.h"
#include "alu.h"
/* ----%< CUT HERE >%---- */

/* **************************** Static Functions **************************** */

static bool
room(char **err)
{
  if (rsidx >= stack_size - 1)
  {
```

```
/* **************************** Static Functions **************************** */

static bool
room(char **err)
{
  if (rsidx >= stack_size - 1)
  {
```

(actually deletes 1 line extra, fixed later)

Keep lines before 1st #include

take 1: Add a “KEEP TO HERE” comment

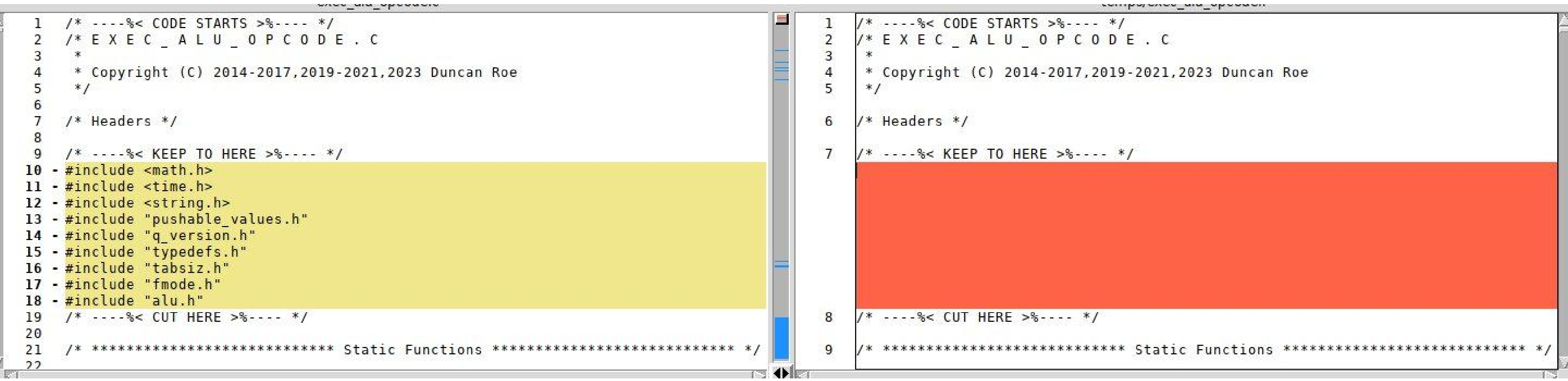


```
/* E X E C _ A L U _ O P C O D E . C
 *
 * Copyright (C) 2014-2017,2019-2021,2023 Duncan Roe
 */

/* Headers */

/* ----%< KEEP TO HERE >%---- */
- #include <math.h>
- #include <time.h>
- #include <string.h>
- #include "pushable_values.h"
- #include "q_version.h"
- #include "typedefs.h"
- #include "tabsiz.h"
- #include "fmode.h"
- #include "alu.h"
/* ----%< CUT HERE >%---- */

/* **************************** Static Functions **************************** */

```

```
+ #define __STDC_IEC_559__ 1
+ #define __STDC_IEC_559_COMPLEX__ 1
+ /* wchar_t uses Unicode 10.0.0.  Version 10.0 of the Unicode Standard is
+    synchronized with ISO/IEC 10646:2017, fifth edition, plus
+    the following additions from Amendment 1 to the fifth edition:
+    - 56 emoji characters
+    - 285 hentaigana
+    - 3 additional Zanabazar Square characters */
+ #define __STDC_ISO_10646__ 201706L
/* E X E C _ A L U _ O P C O D E . C
 *
 * Copyright (C) 2014-2017,2019-2021,2023 Duncan Roe
 */

/* Headers */

/* ----%< KEEP TO HERE >%---- */
/* ----%< CUT HERE >%---- */

/* **************************** Static Functions **************************** */
```

Keep lines before 1st #include

take 2: Insert a “CODE STARTS” comment

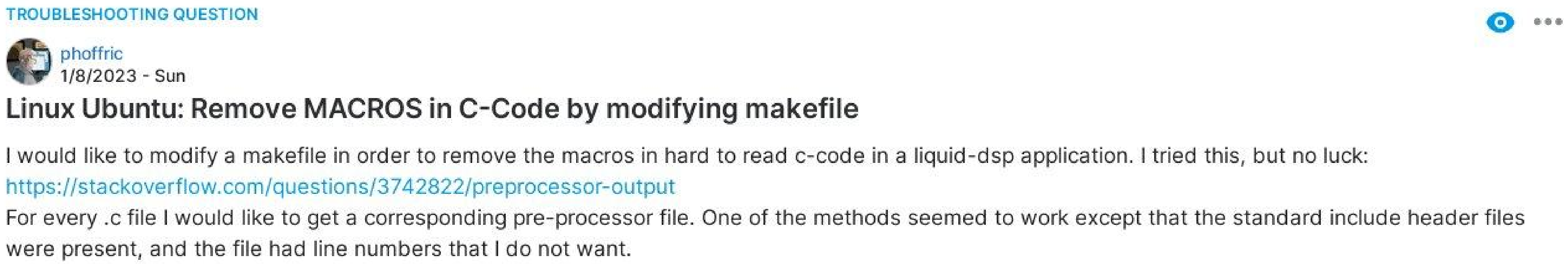


```
/* ----%< CODE STARTS >%---- */
/* E X E C _ A L U _ O P C O D E . C
 *
 * Copyright (C) 2014-2017,2019-2021,2023 Duncan Roe
 */

/* Headers */

/* ----%< KEEP TO HERE >%---- */
- #include <math.h>
- #include <time.h>
- #include <string.h>
- #include "pushable_values.h"
- #include "q_version.h"
- #include "typedefs.h"
- #include "tabsiz.h"
- #include "fmode.h"
- #include "alu.h"
/* ----%< CUT HERE >%---- */

/* *************************** Static Functions *************************** */
```

```
/* ----%< CODE STARTS >%---- */
/* E X E C _ A L U _ O P C O D E . C
 *
 * Copyright (C) 2014-2017,2019-2021,2023 Duncan Roe
 */

/* Headers */

/* ----%< KEEP TO HERE >%---- */

/* ----%< CUT HERE >%---- */

/* *************************** Static Functions *************************** */
```

looks good

Original Experts Exchange question



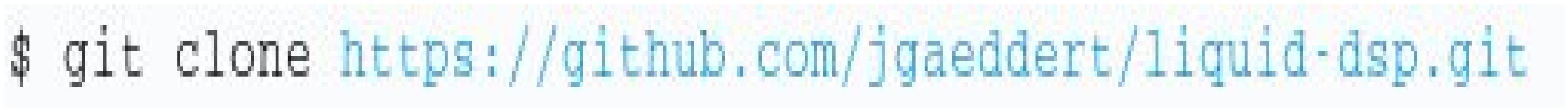
TROUBLESHOOTING QUESTION

phoffric
1/8/2023 - Sun

Linux Ubuntu: Remove MACROS in C-Code by modifying makefile

I would like to modify a makefile in order to remove the macros in hard to read c-code in a liquid-dsp application. I tried this, but no luck:
https://stackoverflow.com/questions/3742822/preprocessor-output
For every .c file I would like to get a corresponding pre-processor file. One of the methods seemed to work except that the standard include header files were present, and the file had line numbers that I do not want.

slightly further on...



```
$ git clone https://github.com/jgaeddert/liquid-dsp.git
```

Did that. First problem: if the last #include is #ifdef'd out, so is immediately following "CUT HERE" comment.
Also discovered that some **.c** files #include other **.c** files

Need better marker comments

- 3 components:

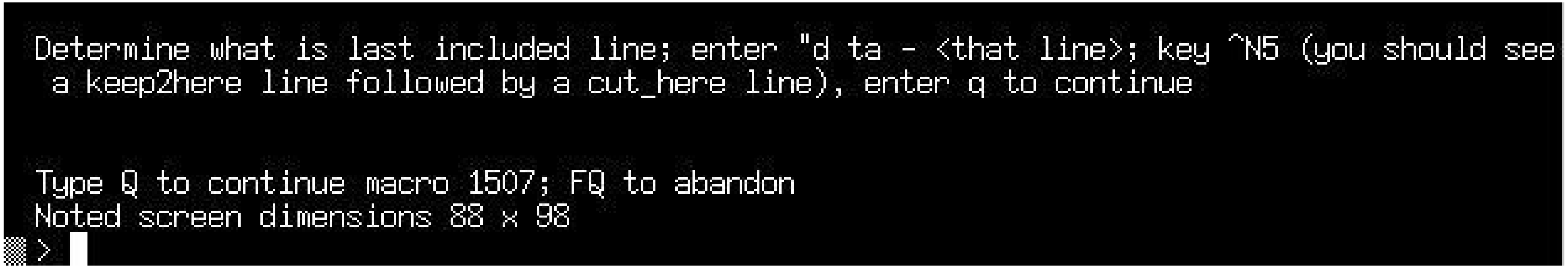
1. *Token*, to easily find next marker. Must not occur in original source
2. *Marker type*, single word (no spaces) e.g. **KEEP2HERE**
3. *Source Path*, originally so **diff** doesn't get false matches. Turns out to be useful for other reasons as well.

Example:

```
/* >%---- CODE_STARTS exec_alu_opcode.c */
```

Invite Manual Edit

You see this



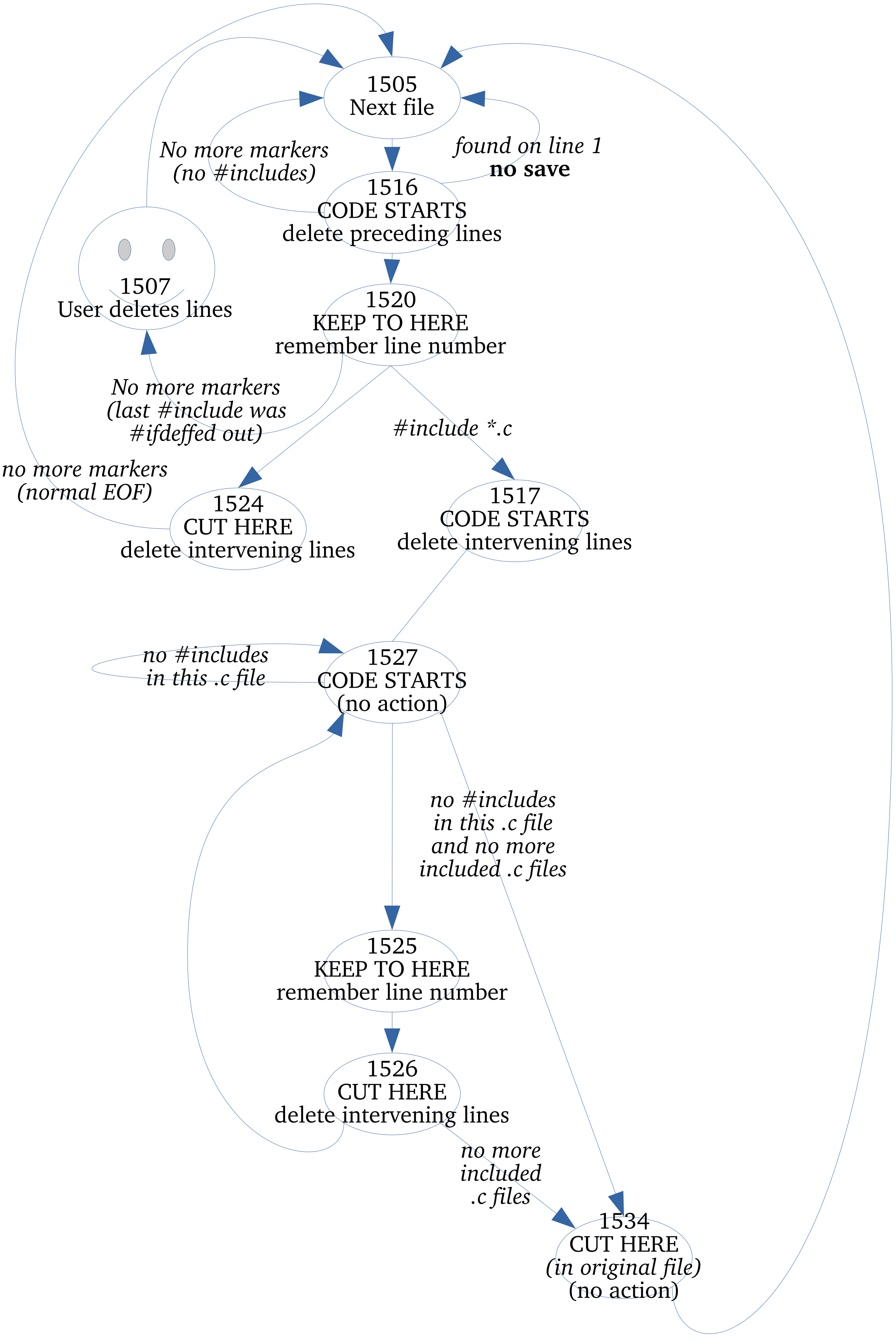
```
Determine what is last included line; enter "d ta - <that line>; key ^N5 (you should see
 a keep2here line followed by a cut_here line), enter q to continue


Type Q to continue macro 1507; FQ to abandon
Noted screen dimensions 88 x 98
▩ > █
```

(quick live demo)

Need more logic to deal with included **.c** files

- Detect already-processed files (for development)
- May get `CUT_HERE` or `CODE_STARTS` after `KEEP2HERE`
- And so on ...

^N3:- process .i files state diagram (sort-of)



1505 Next file
No more markers (no #includes)
found on line 1 **no save**
1516 CODE STARTS delete preceding lines
1507 User deletes lines
1520 KEEP TO HERE remember line number
No more markers (last #include was #ifdeffed out)
#include *.c
no more markers (normal EOF)
1524 CUT HERE delete intervening lines
1517 CODE STARTS delete intervening lines
no #includes in this .c file
1527 CODE STARTS (no action)
no #includes in this .c file and no more included .c files
1525 KEEP TO HERE remember line number
1526 CUT HERE delete intervening lines
no more included .c files
1534 CUT HERE (in original file) (no action)

Challenges from phoffric

- *All user .h files get their macros expanded to form .ih files.*
- *A mirror folder structure matching the liquid-dsp folder structure is defined and the .i and the .ih files are just named .c and .h (but having no macros).*

Challenges from phoffric (continued)

- *Difficult (and negates some of the above goals): Create separate sibling folders if shared files (.h or proto) take on different content as a result of the macros being expanded differently as a result of some earlier macro expansion.*

Using Q editor macros to generate preprocessed Source Code without headers

Part 3: create a shadow tree to diff against

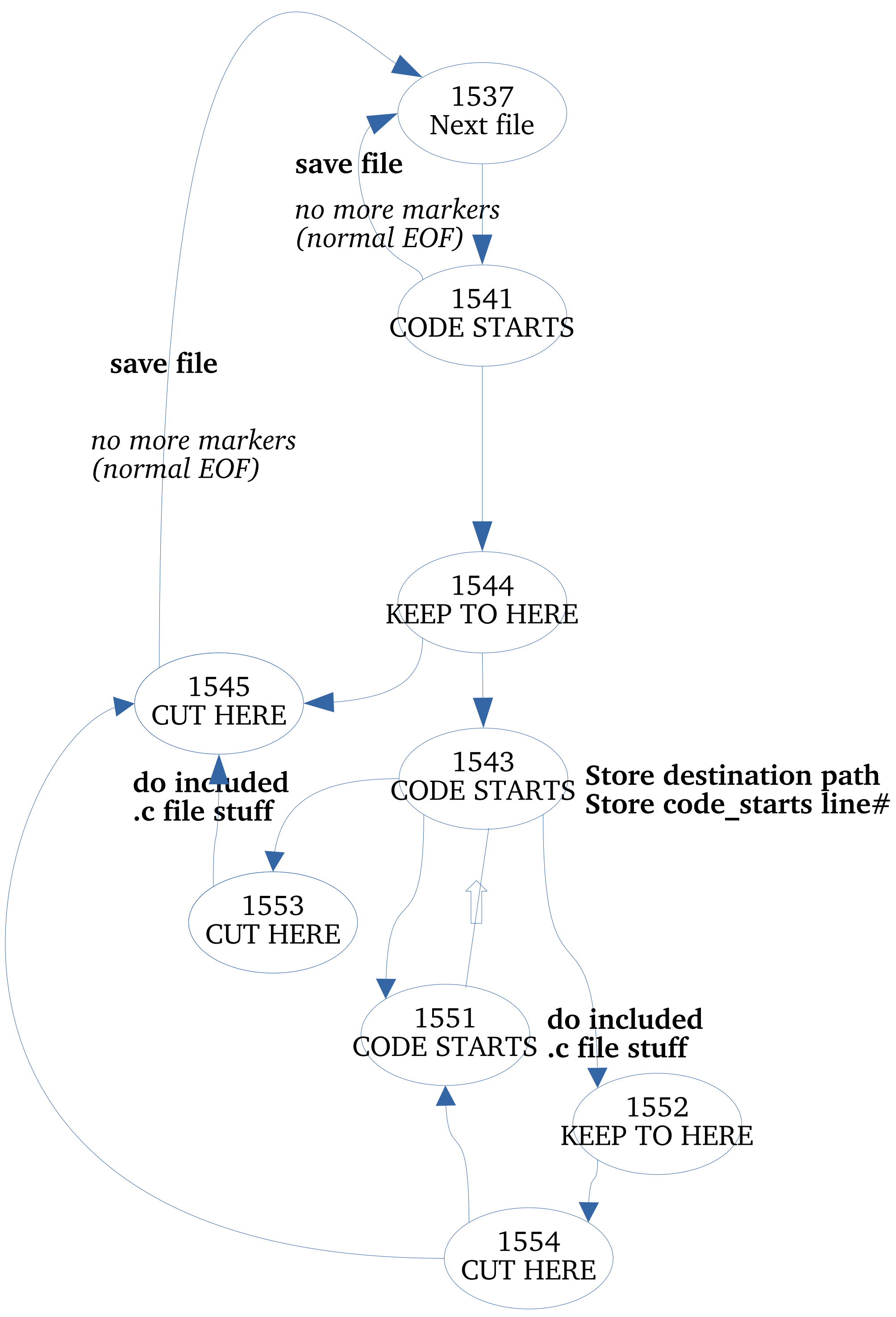
Make a “Master Shadow” tree

- New macro ^N+ creates the shadow tree:

```
rm -rf shadow; q -oiu,cpp.qm^J^N+ temps/*.i
```

- Each included .c file (***proto.c file***) is written out to a file named *`<proto.c file>.<basename including file .c>`*. Proto.c file is removed from including file.
- With this naming scheme, can create a shadow tree with any combination of expanded proto.c files.

^N+:- make shadow directory from .i files



Included .c file stuff

If included file already exists, rename it with backup suffix
Write out included file and delete it from including file
If we made a backup, compare and delete if same else ask user to take action

It’s all shell scripting from now on

- Eventual plan is to have proto.c files as symlinks so can tell from `ls -l` where they came from.
- But first, check for proto.c files that are only #included once. These can be safely **mv**’d into place.

Commands to make *shadow1/*

shadow1/ has a number of proto.c files in place because they never change (except perhaps with different ./configure options, but we’re not going there). Will use *shadow1/* as a template to build individual shadow trees.

```
find shadow -type f ! -name '*.c'|rev|cut -d. -f2-|cut -d/ -
f1|rev|sort -u >p1
cat p1|glb -v '\.c$'|rev|cut -d. -f2-|cut -d/ -f1|rev|sort -
u>p1a
cat p1|glb '\.c$' >p2
cat p1a >>p2
(cd shadow; for i in $(cat ../p2);do if [ $(l $i|wc -l) -eq
1 ]; then echo $i;fi;done) >u
rm -rf shadow1
cp -a shadow shadow1
(cd shadow1; for i in $(cat ../u); do j=$(l $i); (cd $(dirname
$j); mv -iv $(basename $j) $i) done)
```

Create sample tree *sy1/*

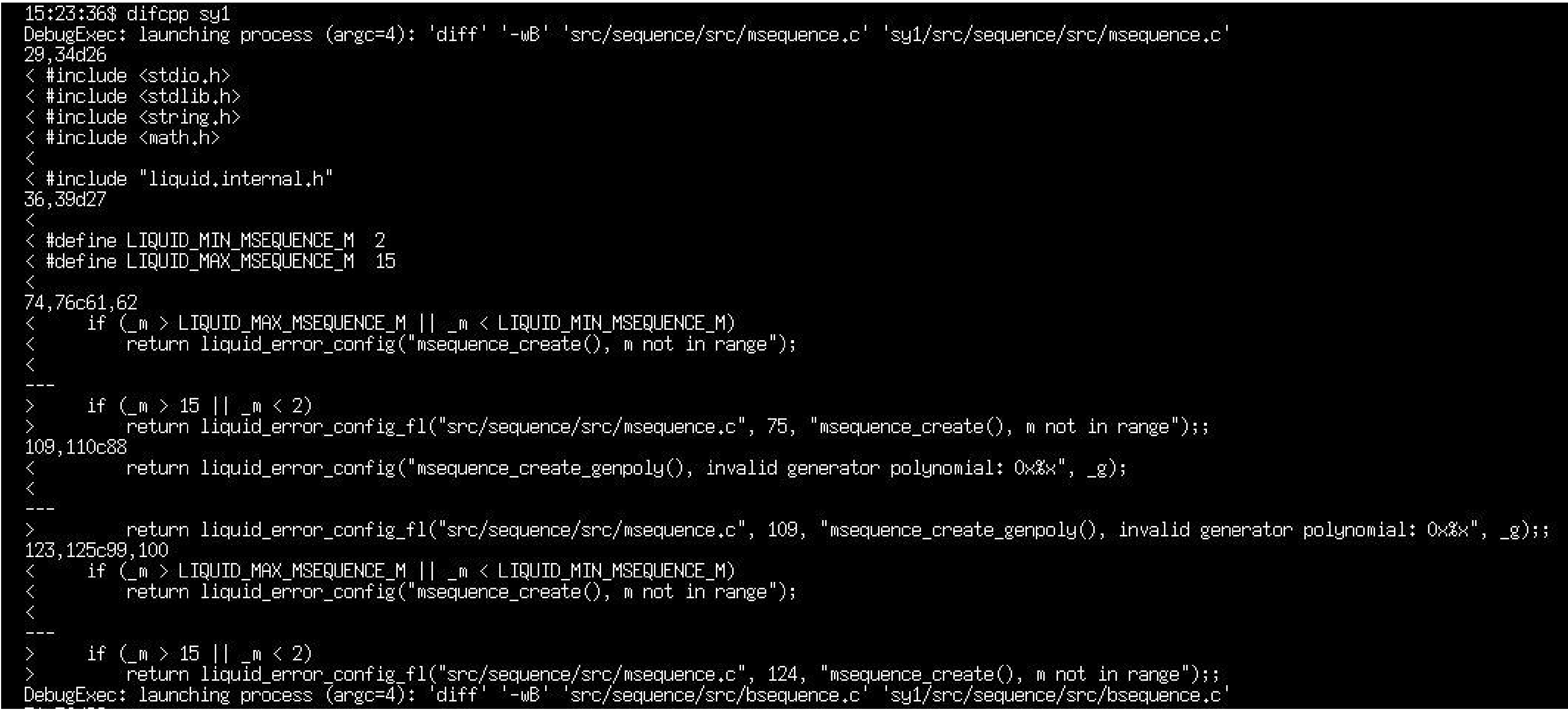
- Function to create individaul .proto.c files:
  ```
  vm(){ (while [ $# -gt 0 ]; do (cd $(dirname $1);ln -s $(basename $1) $
  (basename $(echo ${1/.c.//}|rev|cut -d/ -f2-|rev).c)); shift; done); }
  ```
  VM() can be used with wildcards, e.g.
  ```
  vm src/buffer/src/*.proto.c.bufferf
  ```
  which picks up *cbuffer.proto.c*, *wdelay.proto.c* & *window.proto.c.*
- Function to compare created tree with original:
  ```
  difcpp(){ find -D exec src -type d \( -name tests -o -name bench \) -prune -o
  -name '*.c' -exec diff -wB {} $1/{} \; 2>&1|glb -v "^DebugExec: process"|k; }
  ```

Commands to populte *sy1/*

```
rm -rf sy1; cp -a shadow1 sy1; cd sy1
vm src/agc/src/agc.proto.c.agc_crcf
vm src/fft/src/*.proto.c.spgramcf
vm src/filter/src/*.c.filter_rrrf
vm src/framing/src/*sync.proto.c.*sync_cccf
vm src/buffer/src/*.proto.c.buffercf
vm src/equalization/src/*.proto.c.equalizer_rrrf
vm src/multichannel/src/firpfbch.proto.c.firpfbch_crcf
vm src/matrix/src/*.c.matrixc
vm src/math/src/poly.*.proto.c.polyf
vm src/quantization/src/quantizer.proto.c.quantizercf
vm src/matrix/src/smatrix.proto.c.smatrixi
vm src/vector/src/vector_add.proto.c.vectorcf_add.port
vm src/vector/src/vector_mul.proto.c.vectorf_mul.port
vm src/vector/src/vector_norm.proto.c.vectorcf_norm.port
vm src/vector/src
cd ..
```

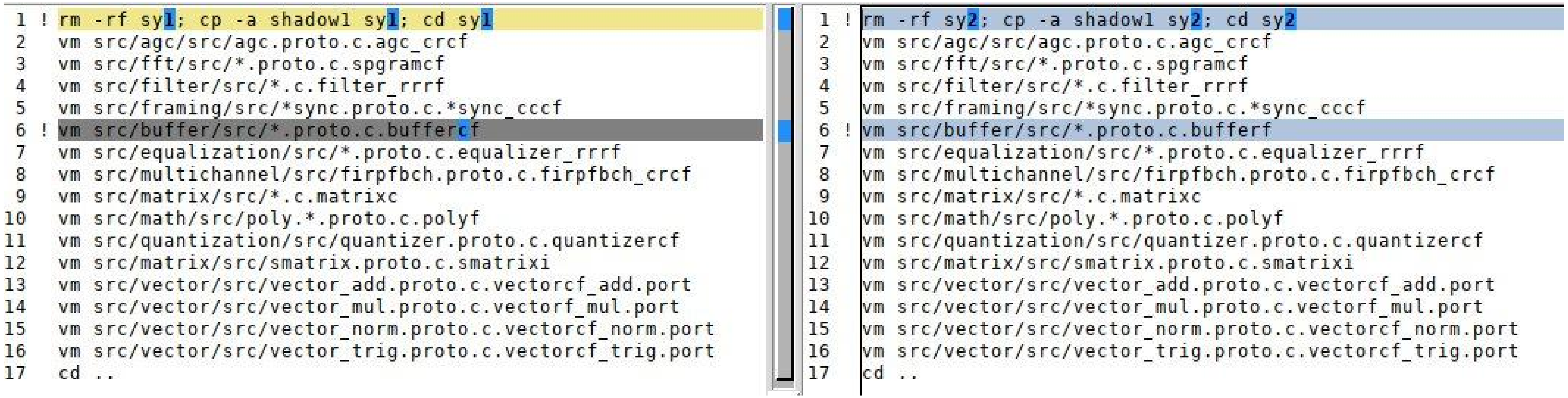
Compare w/original source

- There is a *DebugExec* line between each file



```
15:23:36$ difcpp sy1
DebugExec: launching process (argc=4): 'diff' '-wB' 'src/sequence/src/msequence.c' 'sy1/src/sequence/src/msequence.c'
29,34d26
< #include <stdio.h>
< #include <stdlib.h>
< #include <string.h>
< #include <math.h>
<
< #include "liquid.internal.h"
36,39d27
<
< #define LIQUID_MIN_MSEQUENCE_M  2
< #define LIQUID_MAX_MSEQUENCE_M  15
<
74,76c61,62
<     if (_m > LIQUID_MAX_MSEQUENCE_M || _m < LIQUID_MIN_MSEQUENCE_M)
<         return liquid_error_config("msequence_create(), m not in range");
<
---
>     if (_m > 15 || _m < 2)
>         return liquid_error_config_fl("src/sequence/src/msequence.c", 75, "msequence_create(), m not in range");;
109,110c88
<         return liquid_error_config("msequence_create_genpoly(), invalid generator polynomial: 0x%x", _g);
<
---
>         return liquid_error_config_fl("src/sequence/src/msequence.c", 109, "msequence_create_genpoly(), invalid generator polynomial: 0x%x", _g);;
123,125c99,100
<     if (_m > LIQUID_MAX_MSEQUENCE_M || _m < LIQUID_MIN_MSEQUENCE_M)
<         return liquid_error_config("msequence_create(), m not in range");
<
---
>     if (_m > 15 || _m < 2)
>         return liquid_error_config_fl("src/sequence/src/msequence.c", 124, "msequence_create(), m not in range");;
DebugExec: launching process (argc=4): 'diff' '-wB' 'src/sequence/src/bsequence.c' 'sy1/src/sequence/src/bsequence.c'
```

Make *sy2/* with 1 changed **vm**

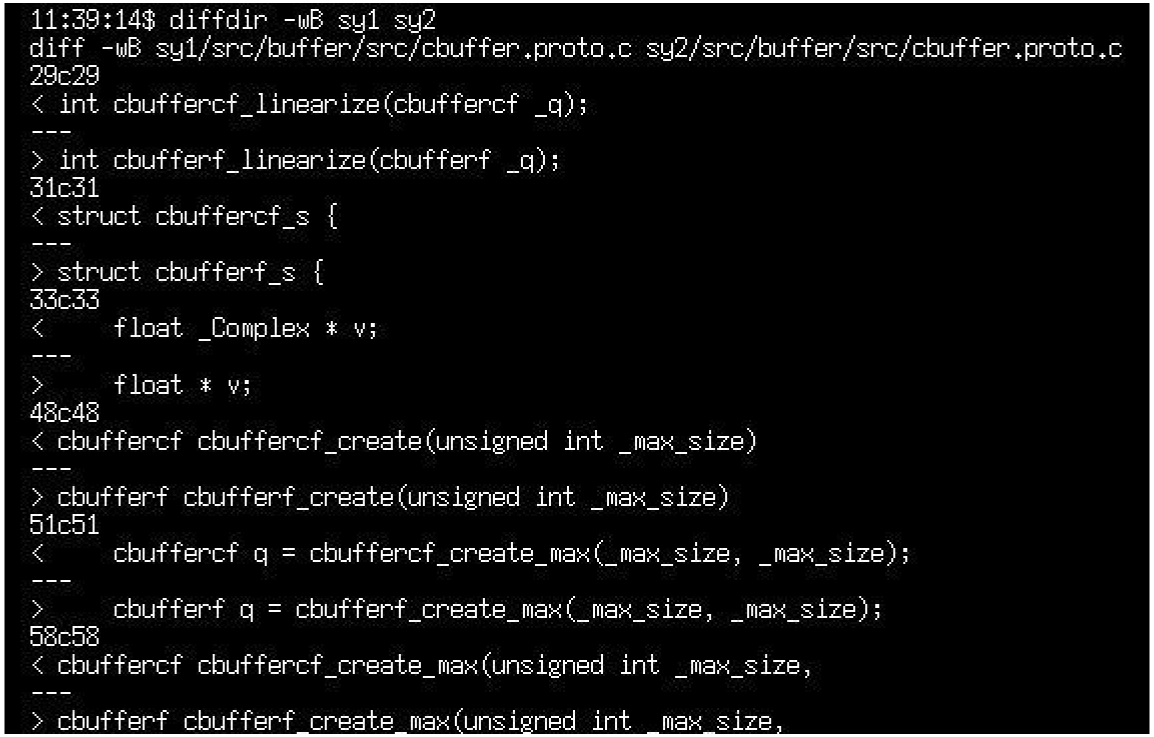


```
! rm -rf sy1; cp -a shadow1 sy1; cd sy1
  vm src/agc/src/agc.proto.c.agc_crcf
  vm src/fft/src/*.proto.c.spgramcf
  vm src/filter/src/*.c.filter_rrrf
  vm src/framing/src/*sync.proto.c.*sync_cccf
! vm src/buffer/src/*.proto.c.buffercf
  vm src/equalization/src/*.proto.c.equalizer_rrrf
  vm src/multichannel/src/firpfbch.proto.c.firpfbch_crcf
  vm src/matrix/src/*.c.matrixc
  vm src/math/src/poly.*.proto.c.polyf
  vm src/quantization/src/quantizer.proto.c.quantizercf
  vm src/matrix/src/smatrix.proto.c.smatrixi
  vm src/vector/src/vector_add.proto.c.vectorcf_add.port
  vm src/vector/src/vector_mul.proto.c.vectorf_mul.port
  vm src/vector/src/vector_norm.proto.c.vectorcf_norm.port
  vm src/vector/src/vector_trig.proto.c.vectorcf_trig.port
  cd ..
```

```
! rm -rf sy2; cp -a shadow1 sy2; cd sy2
  vm src/agc/src/agc.proto.c.agc_crcf
  vm src/fft/src/*.proto.c.spgramcf
  vm src/filter/src/*.c.filter_rrrf
  vm src/framing/src/*sync.proto.c.*sync_cccf
! vm src/buffer/src/*.proto.c.bufferf
  vm src/equalization/src/*.proto.c.equalizer_rrrf
  vm src/multichannel/src/firpfbch.proto.c.firpfbch_crcf
  vm src/matrix/src/*.c.matrixc
  vm src/math/src/poly.*.proto.c.polyf
  vm src/quantization/src/quantizer.proto.c.quantizercf
  vm src/matrix/src/smatrix.proto.c.smatrixi
  vm src/vector/src/vector_add.proto.c.vectorcf_add.port
  vm src/vector/src/vector_mul.proto.c.vectorf_mul.port
  vm src/vector/src/vector_norm.proto.c.vectorcf_norm.port
  vm src/vector/src/vector_trig.proto.c.vectorcf_trig.port
  cd ..
```

Compare *sy1/* and *sy2/*

- *cbuffer.proto.c*, *wdelay.proto.c* & *window.proto.c* are changed.



```
11:39:14$ diffdir -wB sy1 sy2
diff -wB sy1/src/buffer/src/cbuffer.proto.c sy2/src/buffer/src/cbuffer.proto.c
29c29
< int cbuffercf_linearize(cbuffercf _q);
---
> int cbufferf_linearize(cbufferf _q);
31c31
< struct cbuffercf_s {
---
> struct cbufferf_s {
33c33
<     float _Complex * v;
---
>     float * v;
48c48
< cbuffercf cbuffercf_create(unsigned int _max_size)
---
> cbufferf cbufferf_create(unsigned int _max_size)
51c51
<     cbuffercf q = cbuffercf_create_max(_max_size, _max_size);
---
>     cbufferf q = cbufferf_create_max(_max_size, _max_size);
58c58
< cbuffercf cbuffercf_create_max(unsigned int _max_size,
---
> cbufferf cbufferf_create_max(unsigned int _max_size,
```

Resources

- cpp.qm (Q Macro file)
- liquid-dsp (the project with included .c files)

```
glb(){ grep -E --line-buffered "$@"; }
l(){ find . -depth \( -name "*"$1"*" -o -name  ".*"$1"*" \) -print; }
k(){ less "$@"; }
diffdir(){ opts=""; while [ $(echo -- "$1"|cut -c4) = '-' ]; do opts="$opts $1";  shift; done;
  if [ -z "$1" -o -z "$2" ]; then echo "Usage:- $(basename "$0") [diff opts] <dir1> <dir to be compared to
dir1>"; return 1; fi
  find "$1" -type d -exec sh -c "diff $opts \"{}\" \"\$(echo \"{}\" | sed s?^\"$1\"?\"$2\"?)\"" \; 2>&1|
glb -v '^Common subdirectories: '; }
```