

A Catalog of Refactorings to Discipline Preprocessor-Based Annotations

```

if (memcopy(new_fn->name, dirent->name, dirent->name_len) &&
    new_fn->name[dirent->name_len] == 0) {
    return BT_EEXIST;
}

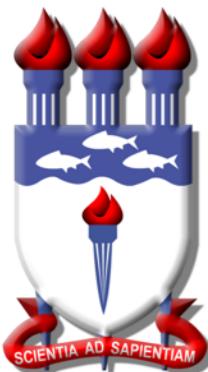
returnwhile_(<(p)>pe{table(filetype))}:
    parent = tp;
}

static int is_dx_dir(struct frame *rb_entry, struct frame, rb_hash):
{
    struct super_block *sb = inode->i_sb;
    if (EXT4_HAS_COMPAT_FEATURE(sb, EXT4_FEATURE_COMPAT_DIRHASH) &&
        (new_fn->hash == frame->hash) &&
        ((ext4_test_inode_flag(inode, EXT4_INODE_INDEX) & new_fn->minor_hash == frame->minor_hash)) &&
        ((inode->i_size > new_fn->next) && frame->next == new_fn)) {
        return 1;
    }
    return 0;
}

static struct dir_private_info *ext4_htree_create_dir_info(struct file *filp,
    if (new_fn->hash < frame->hash)
        p = &(<p>)->rb_left;
    else if (new_fn->hash > frame->hash)
        p = &(<p>)->rb_right;
    if (!p)
        return NULL;
    else if (new_fn->minor_hash < frame->minor_hash)
        p->curr_hash = posn1;
    else if (new_fn->minor_hash > frame->minor_hash)
        p->curr_hash = posn2;
    return p;
}

if (err != ERR_BAD_DIR_HASH) {
    if (err >= ERR_BAD_DIR_HASH) {
        goto out;
    }
    if (info->extra_fname) {
        free_rb_tree_name(ap->root);
        kfree(p);
        goto finished;
    }
    clear_inode_flag(filp->f_path.dentry->i_flags, EXT4_INODE_INDEX);
    info->extra_fname = NULL;
    if (!parent)
        *root = RB_ROOT;
    else if (parent->rb_left == n)
        info->curr_node = rb_first(&info->root);
    else if (parent->rb_right == n)
        parent->rb_right = NULL;
    n = parent;
    if ((info->curr_node) || (filp->f_version != inode->i_version)) {
        map.m_iblk = filp->f_pos >> EXT4_BLOCK_SIZE_BITS;
        map.m_len = 1;
        err = ext4_map_blocks(NULL, inode, &map, 0);
        if (err > 0)
            pgoff_t index = map.m_iblk >>
                (PAGE_CACHE_SHIFT - inode->i_blkbits);
            if (!ra_has_index(&filp->f_ra, index))
                page_cache_sync_readahead(
                    sb->s_bdev->bd_inode->i_mapping, return ret,
                    &filp->f_ra, filp, if (ret == 0) {
                        index, 1);
                    filp->f_pos = ext4_get_htree_eof(filp);
                    filp->f_ra.prev_pos = (loff_t)index << PAGE_CACHE_SIZE;
                    bh = ext4_bread(NULL, inode, map.m_iblk, 0, ver);
                    n = kcalloc(sizeof(struct dir_private_info),
                    parent = rb_parent(n);
                    fname = rb_entry(n, struct frame, rb_
                    while (frame) {
                        struct frame *old = frame;
                        if (call_faddir(filp, dirent, faddir, n, info->extra_fname))
                            kfree(old);
                        frame = rb_next(frame);
                    }
                    if (parent)
                        *root = RB_ROOT;
                    else if (parent->rb_left == n)
                        parent->rb_right = NULL;
                    n = parent;
                    if ((info->curr_node) || (filp->f_version != inode->i_version)) {
                        map.m_iblk = filp->f_pos >> EXT4_BLOCK_SIZE_BITS;
                        map.m_len = 1;
                        free_rb_tree_name(&info->root), EXT4_FT_MAX));
                        filp->f_version = inode->i_version;
                        ret = ext4_htree_fill_tree(filp, info->curr_hash,
                        info->curr_minor_hash, type));
                        info->next_hash;
                    }
                }
            }
        }
    }
}

```



Márcio Ribeiro
<http://www.ic.ufal.br/marcio>
marcio@ic.ufal.br
@marciomribeiro



```
this.dragon = Resources.getImage(Resources.IMG_DRAGON);
```



64kb, without flip



4Mb, without flip



100kb, with flip

```
private int[] xClouds = new int[]{20, 80, 40, 60};
```

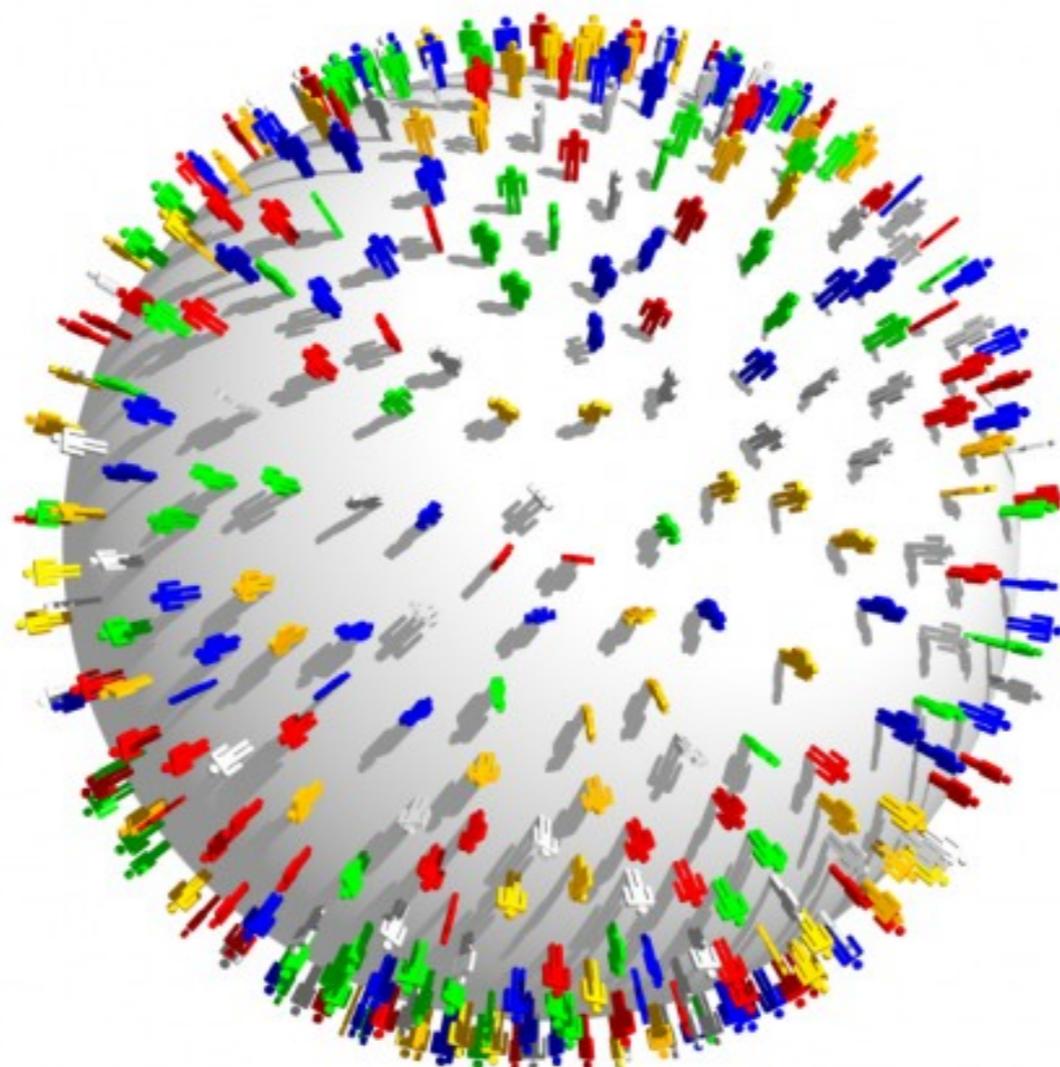
```
private int[] xClouds = new int[]{40, 140, 80, 110};
```

2^n

$n = \{\text{CLOUD}, \text{ RAIN}\}$

Products = [{}, {CLOUD}, {RAIN}, {CLOUD, RAIN}]

33 features



One different product to each **person** in the world

.config - Linux/x86 3.17.3 Kernel Configuration

Linux/x86 3.17.3 Kernel Configuration

Arrow keys navigate the menu. <Enter> selects submenus ---> (or empty submenus ----). Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [*] built-in [] excluded <M> module < > module capable

```
[*] 64-bit kernel
    General setup --->
    [*] Enable loadable module support --->
    [*] Enable the block layer --->
        Processor type and features --->
        Power management and ACPI options --->
        Bus options (PCI etc.) --->
        Executable file formats / Emulations --->
    [*] Networking support --->
        Device Drivers --->
        Firmware Drivers --->
        File systems --->
        Kernel hacking --->
v(+)
```

<**Select**> < Exit > < Help > < Save > < Load >



2

12.000

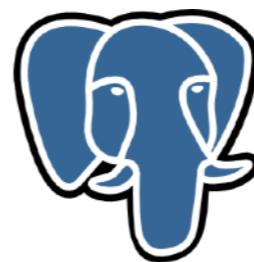


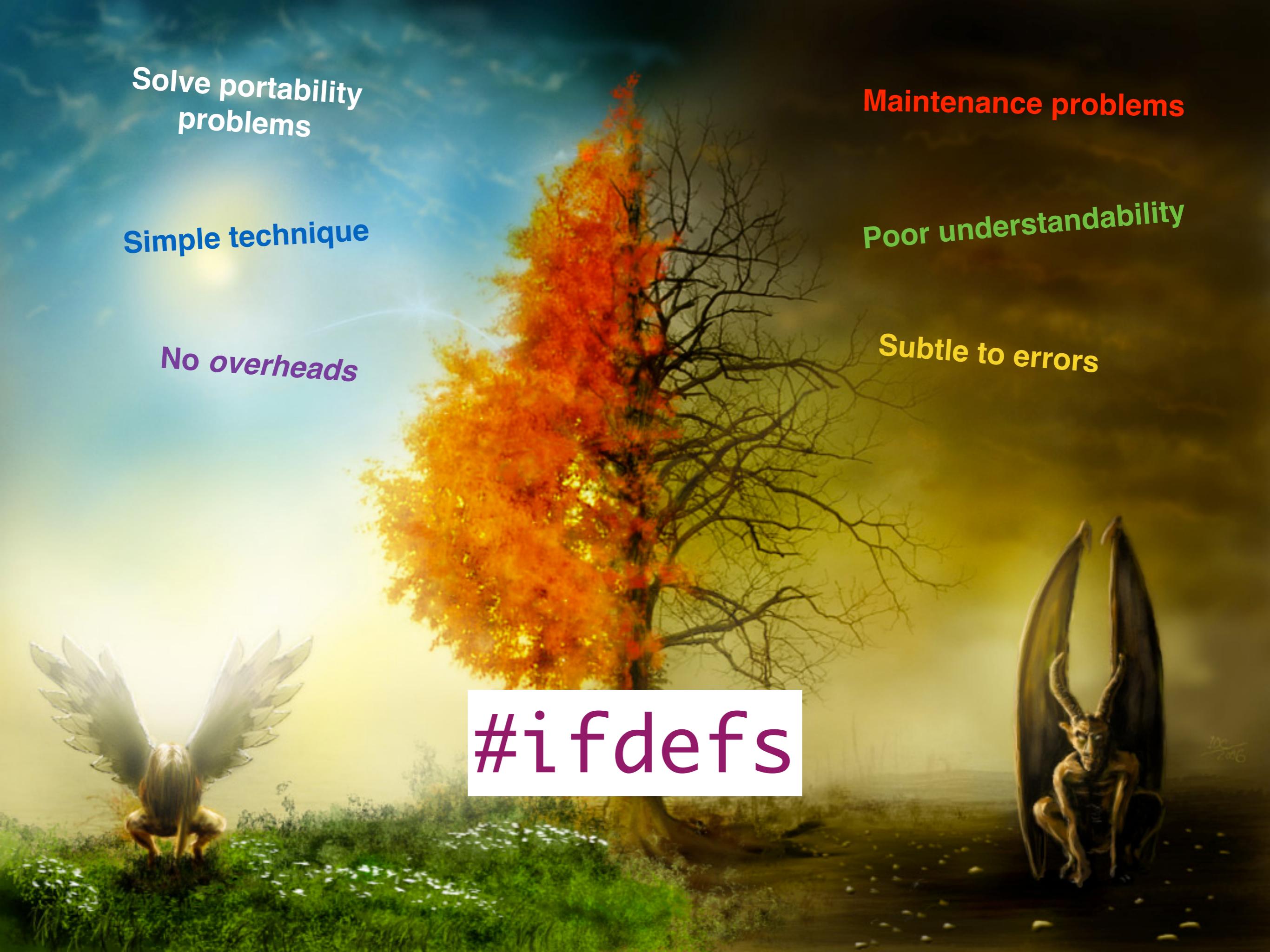
**How to
Implement?**

```
void test() {  
    // code 1  
    #ifdef CLOUD  
        // code 2  
    #endif  
    #ifdef RAIN  
        // code 3  
    #endif  
    // code 4  
}
```

!CLOUD and !RAIN
CLOUD and !RAIN
!CLOUD and RAIN

```
void test() {  
    // code 1  
    #ifdef CLOUD  
        // code 2  
    #endif  
    #ifdef RAIN  
        // code 3  
    #endif  
    // code 4  
}
```





Solve portability
problems

Simple technique

No *overheads*

Maintenance problems

Poor understandability

Subtle to errors

#ifdefs

Undisciplined vs. Disciplined Annotations

```
#if (WIN32)
if ((ready = select(0, NULL, tp))
#else
if ((ready = select(max_fd + 1, NULL, tp))
#endif
    == -1) {
    ngx_log_error(NGX_LOG_ALERT);
    return NGX_ERROR;
    if (ready == -1) {
        err = ngx_socket_errno;
    } else {
        err = 0;
    }
}
```



```
#if (WIN32)
    ready = select(0, NULL, tp);
#else
    ready = select(max_fd + 1, NULL, tp);
#endif
if (ready == -1) {
    ngx_log_error(NGX_LOG_ALERT);
    return NGX_ERROR;
    if (ready == -1) {
        err = ngx_socket_errno;
    } else {
        err = 0;
    }
}
```

Previous
work...

GPCE 2013

ness and response time. Our results indicate that the discipline of annotations has *no influence on program comprehension and maintenance*, neither for correctness nor for performance (in terms of response time). Although we observed some tendencies, they are not supported by our statistical analysis. However, our experiment

```

#if defined(FEAT_XCLIPBOARD) || defined(USE_XSMP) || defined(FEAT_MZSCHEME)
    static int busy = FALSE;

#if defined(HAVE_GETTIMEOFDAY) && defined(HAVE_SYS_TIME_H)
    if (msec > 0 && (
#endif FEAT_XCLIPBOARD
    xterm_Shell != (Widget)0
#endif defined(USE_XSMP) || defined(FEAT_MZSCHEME)
    ||
#endif
#endif
#endif USE_XSMP
    xsmp_icefd != -1
#endif FEAT_MZSCHEME
    ||
#endif
#endif
#endif FEAT_MZSCHEME
    (mzthreads_allowed() && p_mzq > 0)
#endif
))
    gettimeofday(&start_tv, NULL);
#endif
    if (busy)
        return 0;
#endif

```

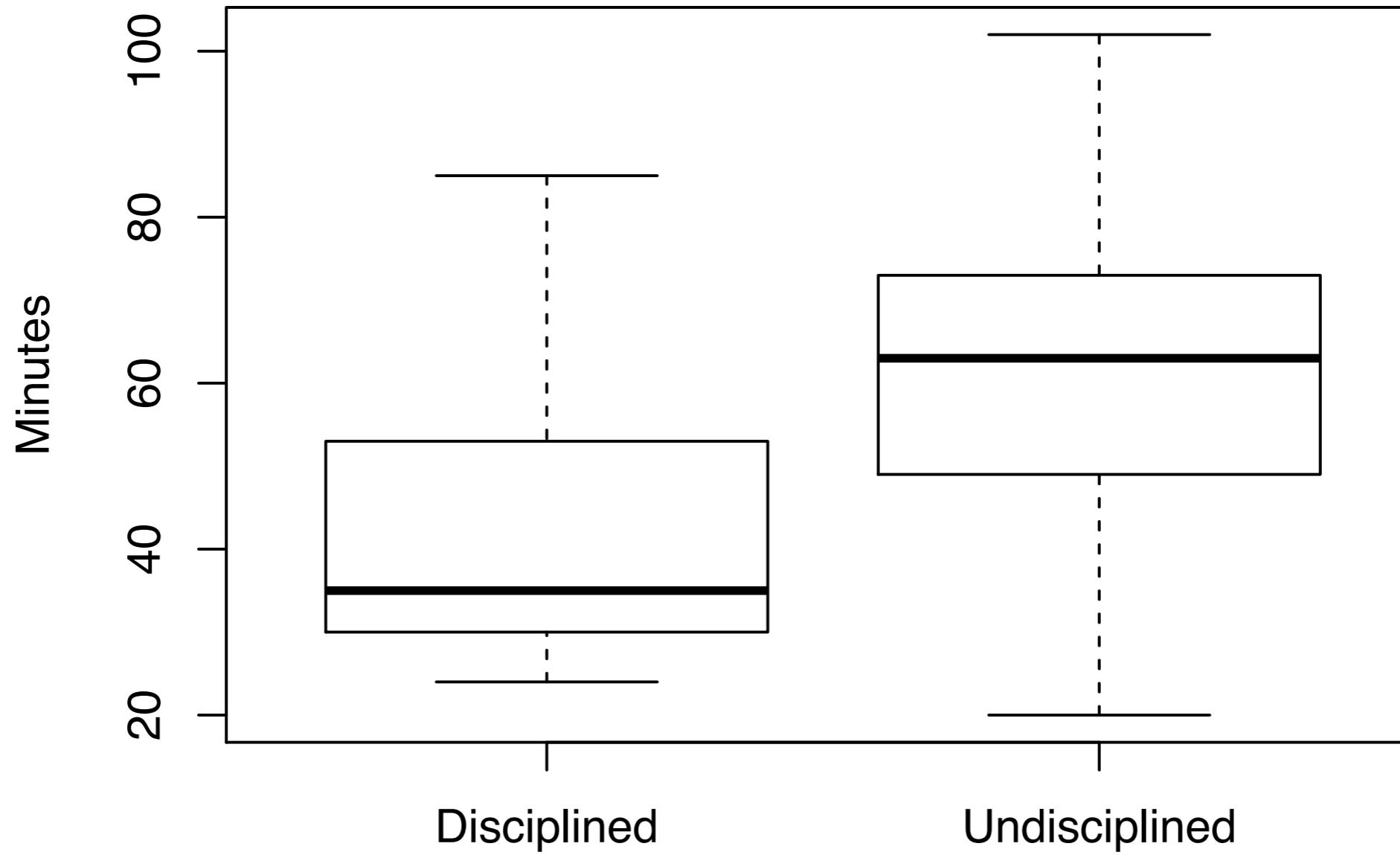


Linux Guidelines

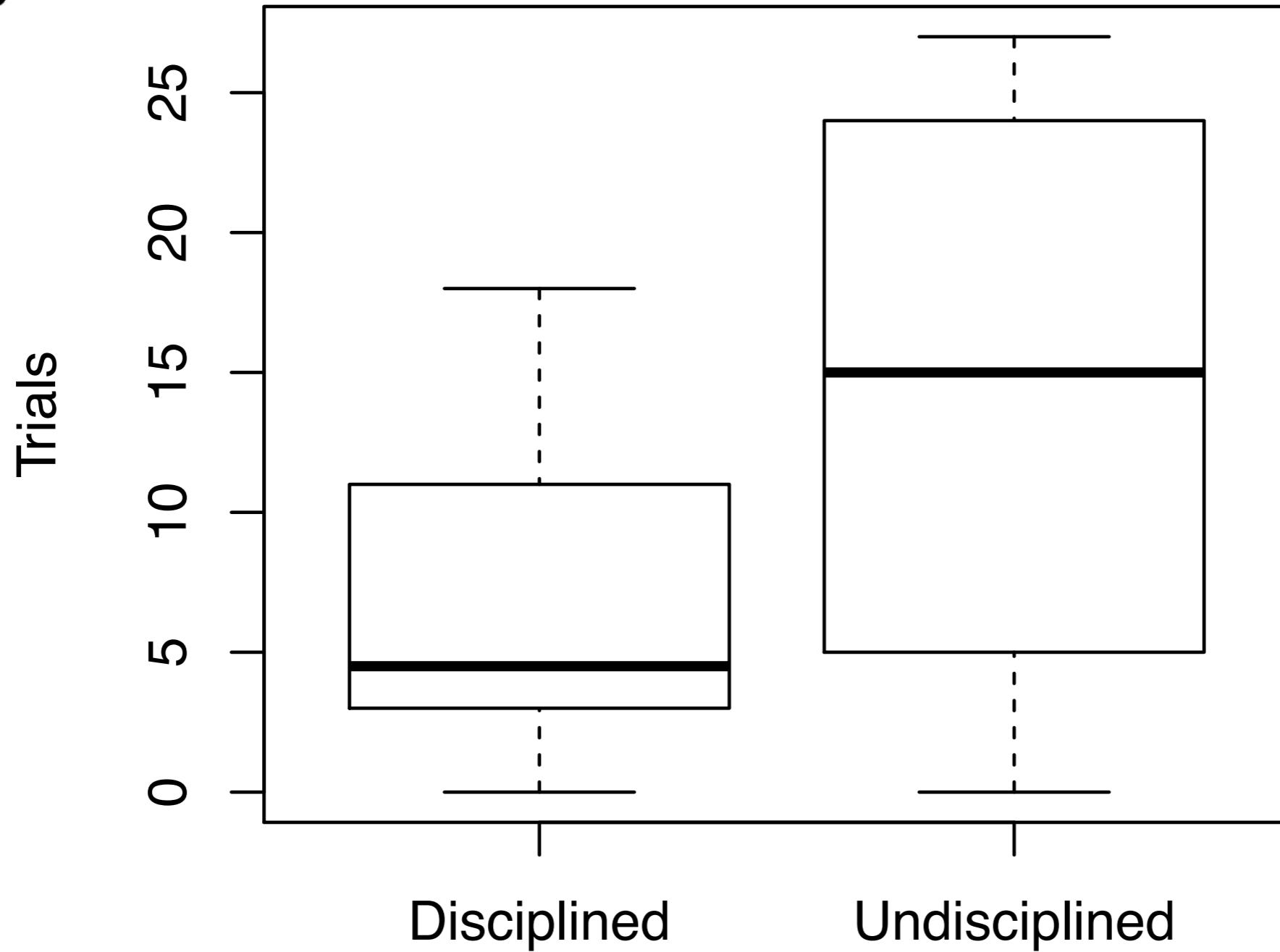


*"Prefer to **compile out entire functions**,
rather than portions of functions or
portions of expressions."*

We found the
opposite



Significant difference!



Significant difference!

Refactoring Undisciplined Annotations

Existing Refactorings...

```
1. #if defined (USE_ISPTS_FLAG)
2. if (result) {
3. #endif
4.     result = open("/dev/ptmx");
5.     if (!result)
6.         strcpy(ttydev);
7. #ifdef USE_ISPTS_FLAG
8. }
9. #endif
```



```
1. #if defined (USE_ISPTS_FLAG)
2. if (result) {
3.     result = open("/dev/ptmx");
4.     if (!result)
5.         strcpy(ttydev);
6. }
7. #else
8.     result = open("/dev/ptmx");
9.     if (!result)
10.        strcpy(ttydev);
11.#endif
```

X Code clone
X Lines of code

Catalogue of Refactorings

Bad Smells	Refactoring
Else inside directives	Else if wrappers
Incomplete if and while statements	Conditional statement wrappers
Case inside directives	Case wrappers
Alternative choice	Alternative statement wrappers
Incomplete else	If statements ending with an else
Incomplete conditions	Conditions
Incomplete commands	Returns and expressions
Incomplete arrays and enums	Data structure definitions
Incomplete function definitions	Function definitions
...	...

14 refactorings



```
1. if (condition1  
2. #ifdef expression1  
3.     && condition2  
4. #endif  
5. ) {  
6.     // Statements..  
7. }
```



```
1. bool test = condition1;  
2. #ifdef expression1  
3.     test = test && condition2;  
4. #endif  
5. if (test) {  
6.     // Statements..  
7. }
```

Precondition: original code is not using the variable test in this scope

- ✓ Code clone
- ✓ Lines of code



```
1. #ifdef expression_1
2.   if (condition_1) {
3. #else
4.   if (condition_2) {
5. #endif
6.     // Lines of code...
7. }
```



```
1. bool test;
2. #ifdef expression_1
3.   test = condition_1;
4. #else
5.   test = condition_2;
6. #endif
7. if (test) {
8.   // Lines of code...
9. }
```

Precondition: original code is not using the variable test in this scope

- ✓ Code clone
- ✓ Lines of code +2



```
1. type function(  
2. #ifdef expression1  
3.     type param_id  
4. #endif  
5. ) {  
6.     // Statements..  
7. }
```



```
1. #ifdef expression1  
2. #define PARAM type param_id  
3. #else  
4. #define PARAM ""  
5. #endif  
6. type function(PARAM) {  
7.     // Statements..  
8. }
```

Precondition: original code does not define the macro PARAM in this scope

- ✓ Code clone
- ✓ Lines of code
- ✗ Code understanding

Evaluating the Refactorings

Frequency of
application
possibilities



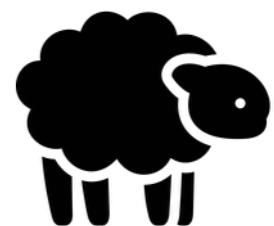
Opinion of
developers



Behavior
preservation

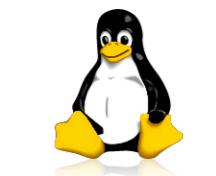


Code
cloning



63
Program Families





1972



905



357



178



270



73



87



23



31



19

7K application possibilities

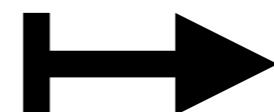




A



```
mfp = open(mf  
#ifdef TS  
    , (mode_t) 0600  
#else  
    , IR | IW  
#endif  
);
```

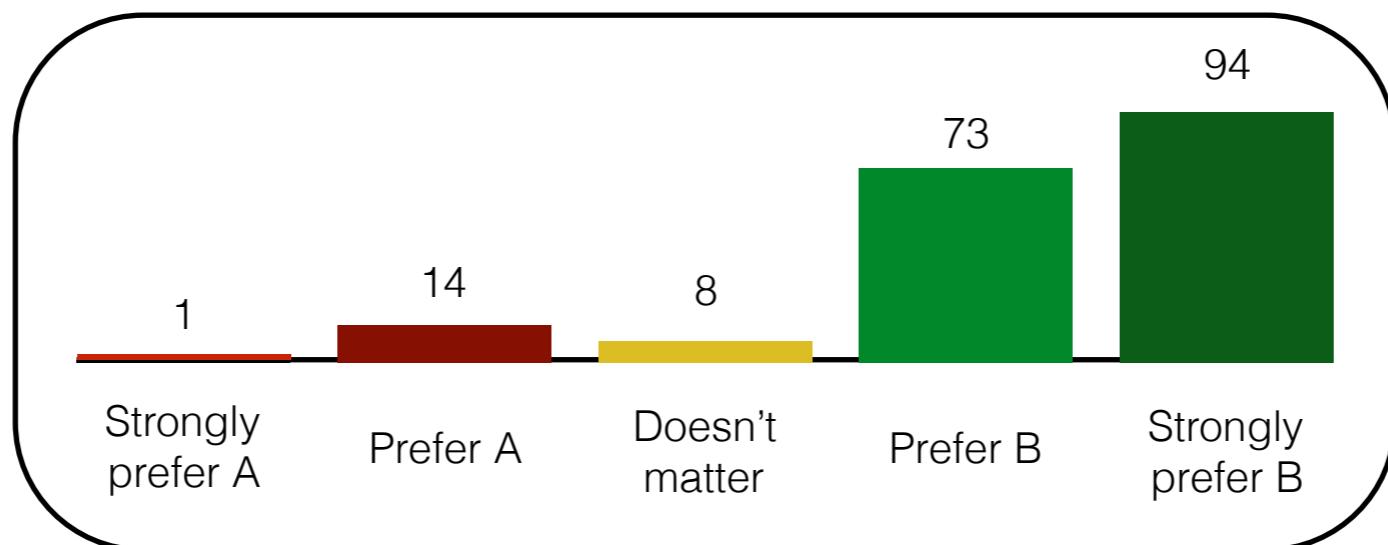


B

```
#ifdef TS  
mfp = open(mf, (mode_t) 0600);  
#else  
mfp = open(mf_name, IR | IW);  
#endif
```

7%

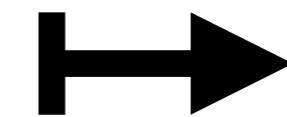
87%





A

```
bool exec = (bit < 8);
#ifndef TS
exec = exec && (r != NULL);
#endif
if (exec) {
    // STMTS
}
```

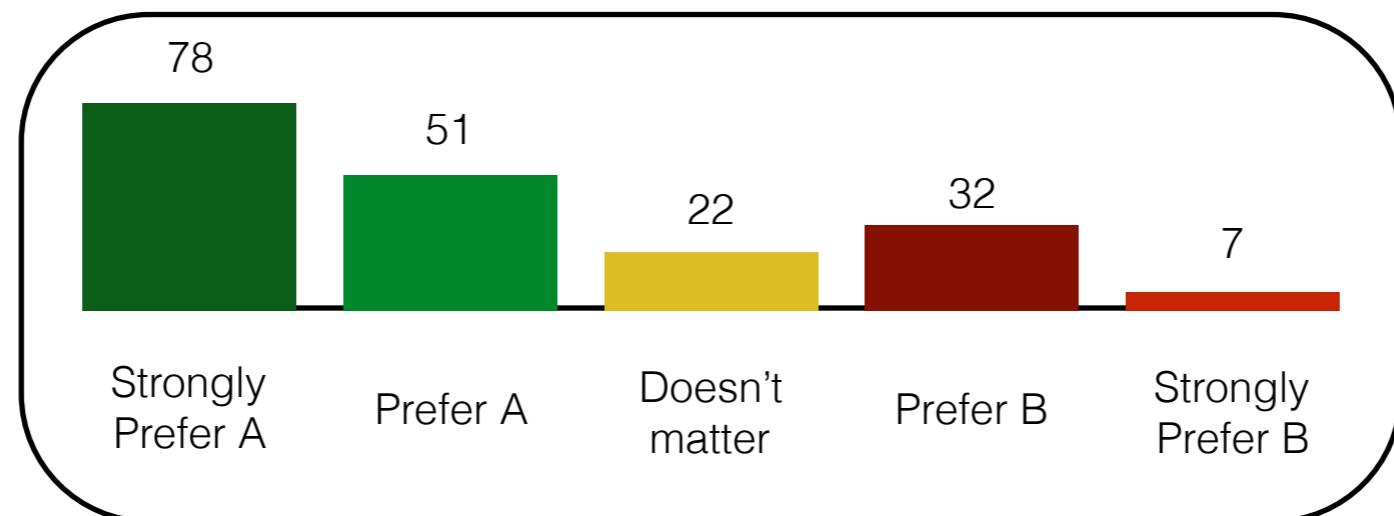


B

```
if (bit < 8
#ifndef TS
    && r != NULL
#endif
) {
    // STMTS
}
```

67%

20%





A

```
#ifdef TS  
#define PARAM Ct server,  
#else  
#define PARAM  
#endif  
void msgNetbeans(PARAM XT client) {  
    // STMTS  
}
```

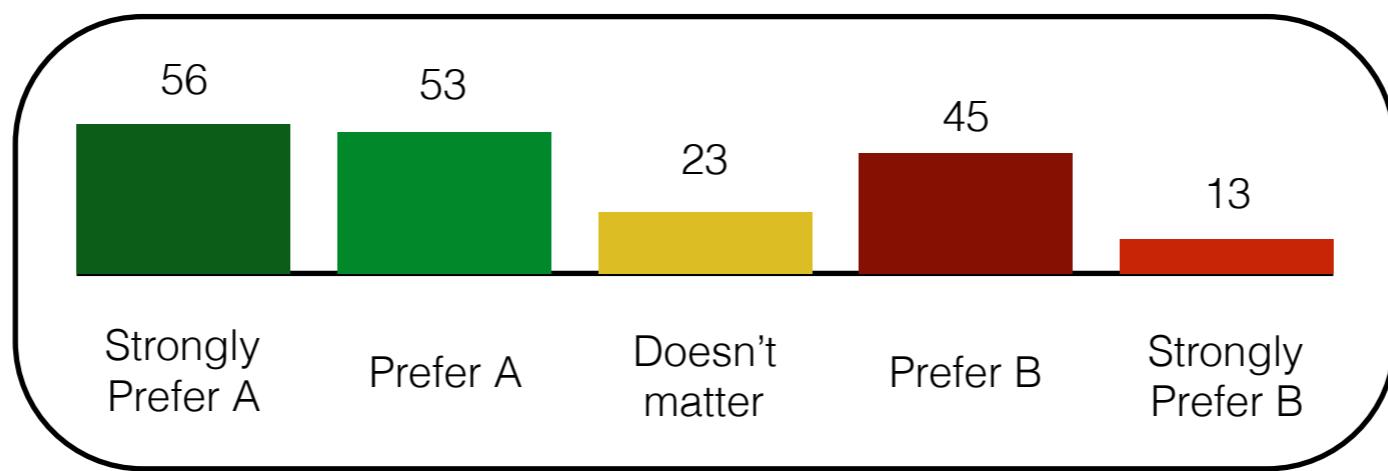


B

```
void msgNetbeans(  
#ifdef TS  
Ct server,  
#endif  
XT client) {  
    // STMTS  
}
```

57%

30%





We received positive feedback from developers when submitting pull requests to discipline undisciplined annotations



```
1. #ifdef expression_1
2.   if (condition_1)
3. #endif
4. {
5.     // Lines of code..
6. }
```

```
1. bool test = 1;
2. #ifdef expression_1
3.   test = condition_1;
4. #endif
5. if (test) {
6.     // Lines of code..
7. }
```

```
1. if ( condition_1
2. #ifdef expression_1
3.   && condition_2
4. #endif
5. ) {
6.     // Lines of code..
7. }
```

```
1. bool test = condition_1;
2. #ifdef expression_1
3.   test = test && condition_2;
4. #endif
5. if (test) {
6.     // Lines of code..
7. }
```

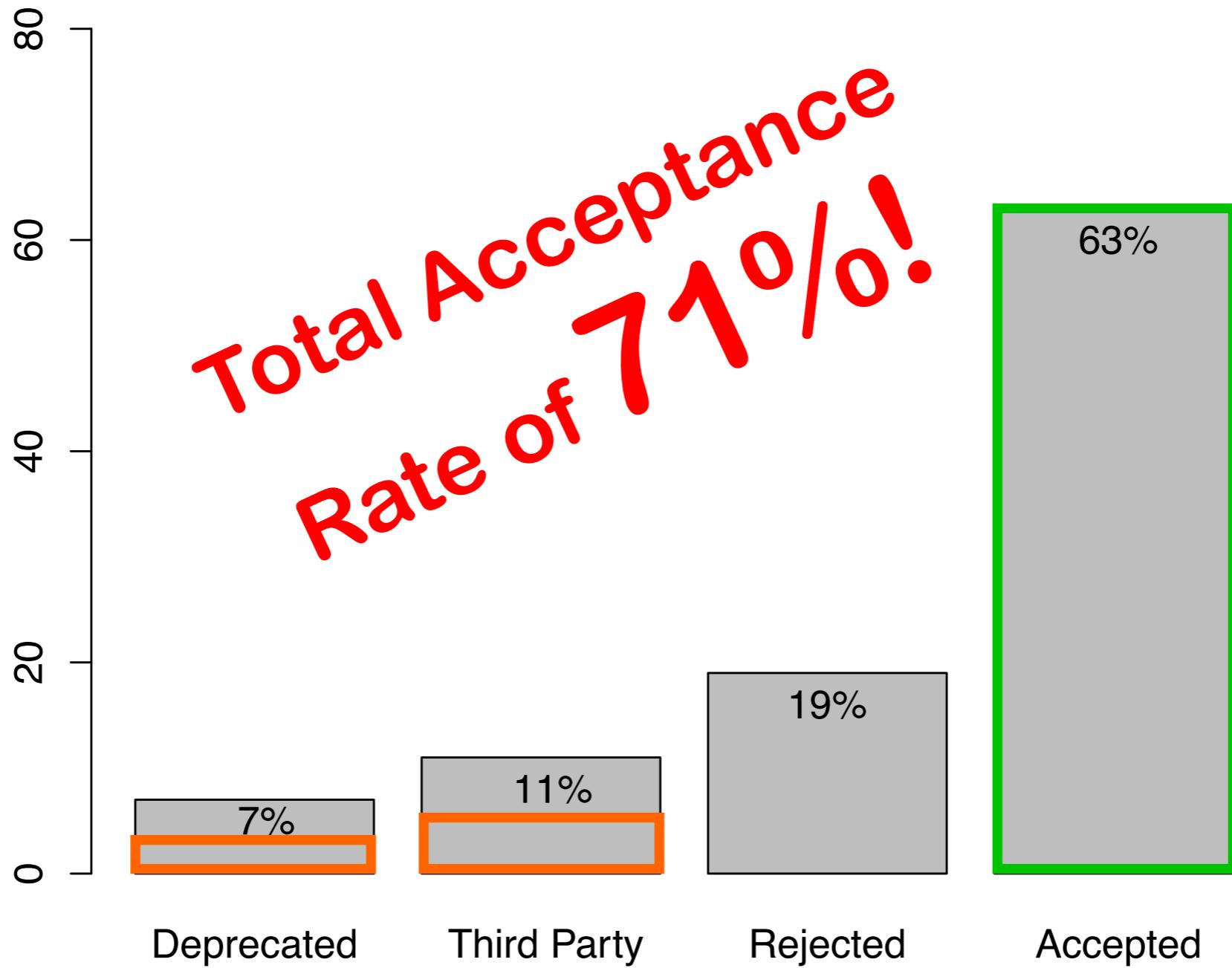
```
1. #ifdef expression_1
2.   if (condition_1){
3. #else
4.   if (condition_2){
5. #endif
6.     // Lines of code..
7. }
```

```
1. #ifdef expression_1
2.   bool test = condition_1;
3. #else
4.   bool test = condition_2;
5. #endif
6. if (test) {
7.     // Lines of code..
8. }
```

110 submitted pull requests. **99** decided!

110 different systems! (one per system!)





Might Accept + Accept



This is not required
Don't like your change

Prefer to save lines of code
No advantages
Improves readability
Third parties code

Not easier to read or understand

Thanks, merged

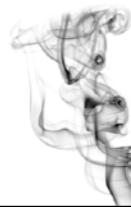
The change is an improvement

Stylistic changes Deprecated
Bad variable name
Breaks the code
That's much better
Sorry, more serious issues to deal with
I want to make as few changes as possible

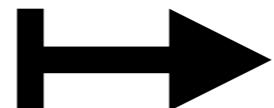




Token sequence



```
call (p1  
#ifdef EXP  
    , p2  
#endif  
);
```



```
#ifdef EXP  
call (p1, p2);  
#else  
call (p1);  
#endif
```

```
#define EXP  
call <> p1 <> p2
```

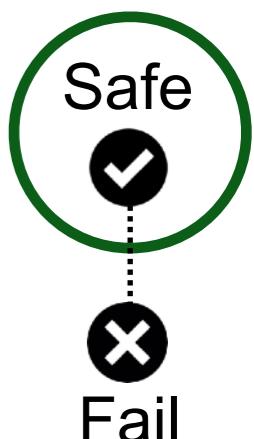
=

```
#define EXP  
call <> p1 <> p2
```

```
#undef EXP  
call <> p1
```

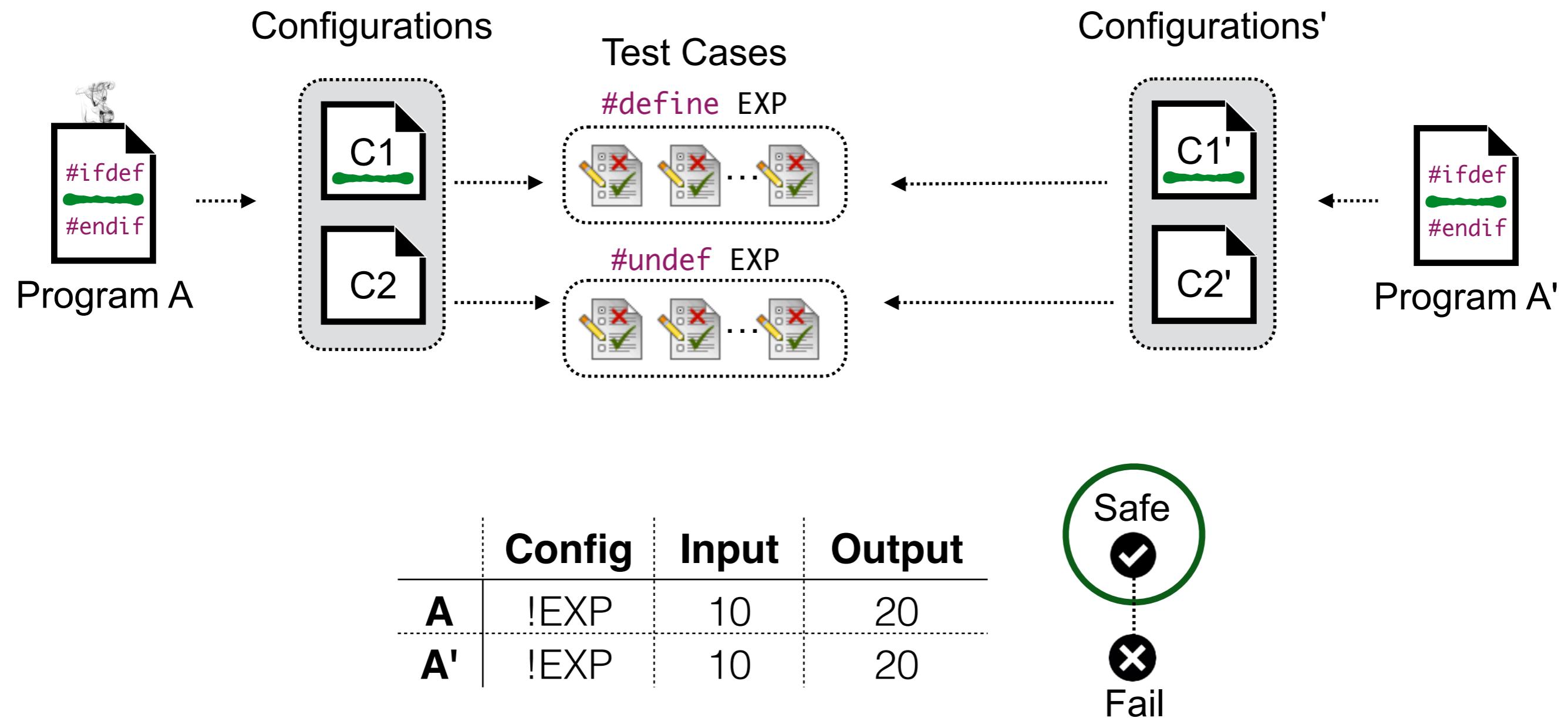
=

```
#undef EXP  
call <> p1
```





Test results

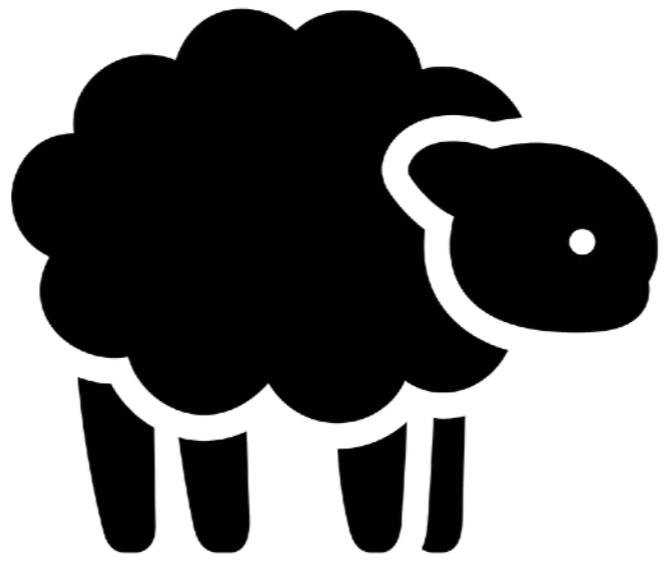


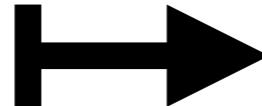
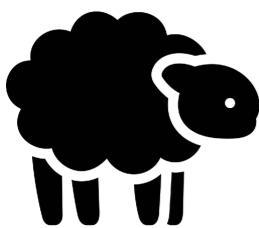


We found and fixed a behavioral change in one refactoring of the catalogue

	R2	R3	R4	R5
Valid programs	7746	7723	14448	6700
Invalid programs	2254	2277	5452	2300
Valid refactorings	7746	7723	14448	6700
Behavioral changes	5	1	5	2

We found and fixed some behavioral changes in the implementation of refactorings



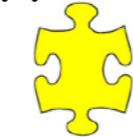


Project	Undisciplined
apache	178
bc	6
dia	31
expat	31
flex	16
fvwm	61
ghostscript	87
gnuchess	2
gzip	19
lighttpd	23
lua	6
mptris	17
Total	477

Clone	LOC	Directives
0	0,18%	2,21%
0	0,12%	0%
0	0,31%	4,06%
0	0,44%	3,87%
0	0,09%	0%
0	0,11%	3,35%
0	0,01%	0,95%
0	0,02%	0%
0	0,64%	4,03%
0	0,08%	1,18%
0	0,12%	3,11%
0	0,78%	3,05%
0	0,04%	2,1%

Colligens

FeatureIDE CppCheck



Eclipse



Core



TypeChef

.c example1.c

Original Source

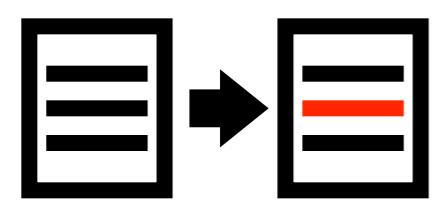
```
void test(){  
  
    int row = 0;  
    int current, first, last = 0;  
  
    row = getRowNumber(current  
    #ifdef A  
        , first, last  
    #endif  
    );  
  
}
```

Refactored Source

```
void test () {  
  
    int row = 0 ;  
    int current,first,last = 0 ;  
  
#if defined(A)  
    row = getRowNumber (current, first, last);  
#endif  
  
#if !defined(A)  
    row = getRowNumber (current);  
#endif  
  
}
```

Buttons: < Back, Next >, Cancel, Finish

Ongoing work...



"If my memory serves correctly, it was just a refactoring to simplify the code and make it a little more understandable."

libpng

"Because using conditional for only part of one high level instruction makes it less understandable and is likely to raise issue over long time maintainance."

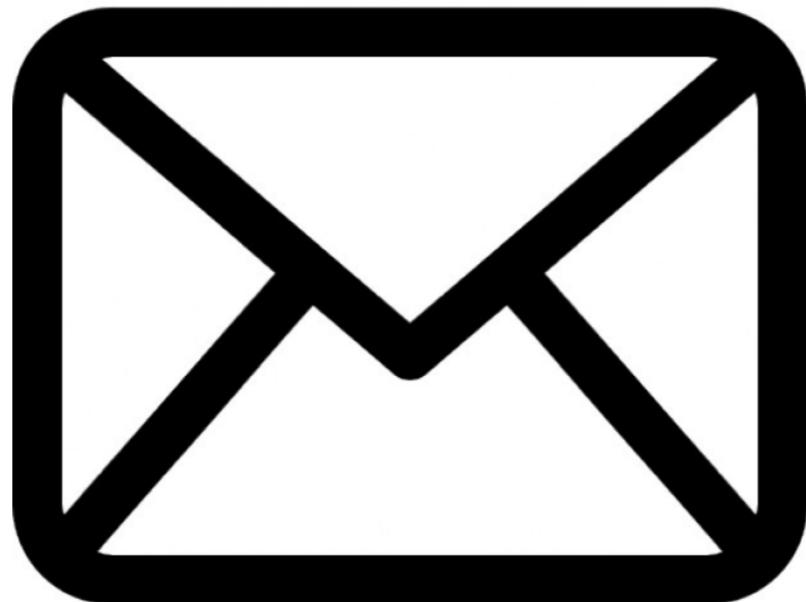


Libxml2

The Discipline of Preprocessor-Based Annotations

Does `#ifdef TAG` n't `#endif` Matter

and let TAG be disabled!



Maceió, Alagoas, Brasil

Takk!



A Catalog of Refactorings to Discipline Preprocessor-Based Annotations

```
if (err <= ERR_BAD_DIR) { ... }
if (memory(new_fn->name, dirent->name, dirent->name_len); ... )
    if (err >= ERR_BAD_DIR) { ... }
    ret = err;
    goto out;
}
new_fn->name[dirent->name_len] = 0;
return DI_UNIONED;
}

return while_(&P){p(table(filetype));
    parent = &p;
}
static int is_dx_dir(struct rb_entry(parent, struct frame, rb_hash);
{
    struct super_block *sb = inode->i_sb;
    if (EXT4_HAS_COMPAT_FEATURE(new_fn->hash == frame->hash) &&
        EXT4_FEATURE_INCOMPAT(ext4_test_inode_feature(new_fn->minor hash == frame->minor hash))
        ((inode->i_size > new_fn->next) || (frame->next == new_fn);
    return 1;
    frame->next = new_fn;
    return 0;
}
return 0;
}

static struct dir_private_info *ext4_htree_create_dir_info(struct file *filp,
    if (new_fn->hash < frame->hash)
        p = &(&p)->rb_left;
    struct dir_private_info *fp(new_fn->hash > frame->hash)
        p = kcalloc(sizeof(struct rb_node), 1);
    if (!p)
        return NULL;
    p->curr_hash = pos->hash;
    p->curr_minor_hash = pos->hash;
    p = &(&p)->rb_right;
}

if (err != ERR_BAD_DIR) {
    if (err >= ERR_BAD_DIR) { ... }
    ret = err;
    goto out;
}
if (info->extra_frame) {
    free_rb_tree_frame(sp->root);
    kfree(p);
}
if (call_faddir(filp, dirent, filaddir, info->extra_frame))
    goto finished;
clear_inode_flag(filp->f_path.dentry->extra_frame = NULL,
    EXT4_INODE_INDEX)goto next_node;
} else if (!info->curr_node)
    stored = 0;
offset = filp->f_pos + (sb->s_blocksize - 1);
while (1) {
    struct ext4_map_blocks map;
    struct buffer_head *bh = NULL;
    if ((!info->curr_node) || ...
        (filp->f_version != inode->i_version)) {
        map.m_iblk = filp->f_pos >> EXT4_BLOCK_SIZE_BITS;
        map.m_len = 1;
        err = ext4_map_blocks(NULL, inode, &map, 0);
        if (err > 0) {
            pgoff_t index = map.m_iblk >>
                (PAGE_CACHE_SHIFT - inode->i_blkbits);
            if (!ra_has_index(filp->f_ra, index))
                page_cache_sync_readahead(
                    sb->s_bdev->bd_inode->i_mapping, return ret;
                    &filp->f_ra, filp, :if (ret == 0) {
                        index, 1);
                        filp->f_pos = ext4_get_htree_eof(filp);
                        bh = ext4_bread(NULL, inode, map.m_iblk, 0, err));
                    p = kcalloc(sizeof(struct dir_private_info),
}

Márcio Ribeiro
http://www.ic.ufal.br/marcio
marcio@ic.ufal.br
@marciomribeiro
```

