特別研究報告

題目

Linux カーネルの著作権表示に関するトークン単位での調査

指導教員
井上 克郎 教授

報告者
田邉　傑士

令和 3 年 2 月 9 日

大阪大学 基礎工学部 情報科学科

令和 2 年度 特別研究報告

Linux カーネルの著作権表示に関するトークン単位での調査

田邉　傑士

内容梗概

ソフトウェアの著作権を持つ者は他人がソフトウェアを変更，再利用，再配布できるかどうか，またどのような条件下であればそれを認めるかを決定する権利などを持つ．OSS と呼ばれるライセンスの下でソースコードが無償で公開されているソフトウェアにおいてもそれは同様である．OSS の著作権を持つ者はライセンスを変更する権利やライセンス違反の利用を行った者に対して著作権侵害で訴える権利を持つ．そのため，OSS の利用を考える者にとって著作権の所在を把握することは訴訟や罰金のリスクを回避するために重要である．

また，OSS はさまざまな国や地域，そしてさまざまな組織に所属する多数の人々が開発している．そのため，特定の企業や団体，個人が独自に保有するソフトウェアとは異なり，一般的な OSS には多くの著作権保持者が存在している．1 つのファイルに複数の著作権保持者が存在することも珍しくない．そのため，著作権表示がなければ利用者が著作権保持者を把握することは困難である．しかし，ソースコードの作成者が必ず著作権表示を追加しているとは限らないため，著作権の所在が示されていないソースコードが存在する恐れがある．

そこで本研究では，OSS における著作権表示の状況を理解するために，OSS の典型的な例である Linux カーネルを研究対象として著作権表示の有無に関する実態をトークン単位で調査した．調査では，著作権表示があるトークンの割合は十分とは言えないこと，すべてのトークンに著作権表示があるファイルの割合は非常に低いということがわかった．また，団体での開発と個人での開発の比較においては，団体での開発の方が個人での開発によりも著作権表示があるトークンの割合がやや高かったものの，大きな違いは見られなかった．また，機能別による著作著作権表示の有無に関する割合には大きな違いが見られた．

主な用語

OSS
著作権
著作権表示

目 次

1　まえがき

OSS(Open Source Software) とは作成者がソースコードを無償で公開しているソフトウェアである．OSS において，著作権を持つ者は OSS のライセンス条項を施行する権利やライセンスを変更できる権利を持つほか，ライセンス違反の利用を行った者に対して著作権侵害で訴える権利を持つ．そのため，OSS を再利用しようと考える企業や団体，個人にとって OSS の著作権保持者を明確にしておくことは重要である．例えば，改変したソースコードの公開がライセンスで義務付けられている場合を考える．この際，自らのプロジェクトで改変したソースコードを公開したくない場合は，OSS の著作権保持者と新たなライセンスの付与を交渉する必要がある．そうしなければ，そのプロジェクトは OSS の著作権保持者によって訴訟されるリスクに晒されることになる．

OSS はさまざまな人々が開発を行っている．例えば，有名な OSS である Linux カーネルは 2019 年末までに 21,074 の異なる人物からコミットを受けている [12]．そしてその各々が自らが追加したソースコード片に対して著作権を主張する事が可能である．このように特定の企業や団体，個人が独自に保有するソフトウェアとは異なり，一般的な OSS では数多くの企業や団体，個人が著作権を有するソースコードが混在している．そのため OSS においてはソースコードの作成者が著作権表示によって著作権保持者を明確にしなければ利用者がソースコードの著作権保持者を特定することは困難である．よって著作権表示は OSS の利用者にとって重要な役割を持つ．しかし，現在 OSS の著作権表示は一部のソースコードの著作権の所在を示していない恐れがある．

そこで OSS における著作権表示の状況を理解し，ソースコードの著作権をより良く管理するための第一歩として本研究では，著作権保持者が異なるソースコードが混在している OSS の一例である Linux カーネルを研究対象とし，その著作権表示の状況を以下の点から調査した．

- RQ1:Linux カーネルにおいて著作権表示があるソースコードの割合はどの程度か？
- RQ2:団体での開発と個人での開発によって著作権表示があるソースコードの割合に違いはあるのか？
- RQ3:機能ごとに分けられたディレクトリによって著作権表示があるソースコードの割合に違いはあるのか？

この論文は次のように構成している．はじめに 2 章で OSS の著作権に関する簡単な背景説明を行った後，3 章で調査内容と調査を行うためのデータセットを構築する方法について説明する．4 章では，実施した調査の結果と考察について述べる．そして 5 章では調査結果

に対する妥当性の脅威とその影響について述べる．最後に 6 章で論文を締めくくり，今後の研究の方向性について概説する．

2　背景

この章ではまず調査内容を理解する上で前提となる著作権と OSS について解説する．そしてそれらが持つ課題を概説する．

2.1　著作権

一般に著作権と呼ばれる権利は支分権と呼ばれるいくつかの権利の集合である．支分権には主に著作物を公表する権利，著作物を頒布する権利，著作物を複製する権利，著作物とその名前を改変されない権利などがある．著作権に関する法律の内容は国によって異なるものの，その多くは世界知的所有権機関 (WIPO) によって設定された共通のガイドラインに基づいている [10]．そして著作権は TRIPS 協定などの国際協定によってソースコードに対しても適用する事が定められており，ソースコードの著作権保持者は上記のような文学的著作物に対する著作権保持者と同様の権利を持つ [9]．ソースコードの著作権はコード片の作成時に発生するため，著作権表示を追加することは義務ではない．また，多くの国において，職務の範囲で作った作品の著作権は作成者が所属する企業や団体に帰属する．

2.2　OSS

OSS(Open Source Software) は作成者がソースコードを無償で公開しているソフトウェアである．一般に作成者は利用者に特定のライセンスの下でソフトウェアの自由な利用を認めている．オープンソースであることの定義はオープンソースの標準化団体によって異なるが，最も有名なオープンソースの標準化団体の 1 つである Open Source Initiative によれば以下のような条件を満たすことをいう [7].

- 無料で再配布可能であること
- プログラムがソースコードを含んでおり，プログラマーがプログラムを変更するのに適した形式であること
- 改変したソフトウェアが元のソフトウェアのライセンスと同じ条件で配布が可能であること
- ソースコードの整合性が取れていること
- 個人または団体に対して差別しないこと
- 特定の分野に対して差別しないこと
- プログラムに付随する権利をそのプログラムが再配布された者全てに等しく認めること

- 特定製品でのみ有効なライセンスを付与しないこと
- 他のソフトウェアの使用を制限するライセンスを付与しないこと
- ライセンスが技術的に中立であること

2.2.1 Linux カーネル

Linux カーネルとは 1991 年に初めてリリースされたオペレーティングシステムの OSS である．今までに 2 万人を超えるコントリビューターが参加し，2020 年 8 月の時点で 100 万件以上のコミットを持つ [12]．OSS の中でも長い歴史と規模を誇り，最も成功した OSS の 1 つであると言える．今回の調査では Linux カーネルを調査対象とした．

2.3 OSS の著作権に関する課題

世界中のほぼすべての場所で，ソースコードは著作権法によって保護されている．それは OSS に対しても同様である．OSS の著作権保持者は著作権を放棄しているわけではなく，ソフトウェアにライセンスを付与し，利用者がライセンスの条項を遵守することを条件に本来著作権保持者のみがもつ支分権の一部の利用を許可しているに過ぎない．よって，OSS の著作権保持者はその OSS に対して公開や頒布をコントロールする権利を保持し続けている．これには，ライセンス条項の施行や変更，ライセンス違反者に対する著作権侵害の訴訟を提起する権利などが該当する．したがって OSS の利用においてライセンスの条項を無視または違反した場合は著作権保持者からの訴訟や罰金のリスクを負うことになる．例えば，過去に Linux カーネルの開発者の 1 人である Patrick McHardy は Linux カーネルにおける自身の著作権を侵害されていると主張して Linux カーネルを利用した一部の企業を訴え，金銭を要求しており [8]，この事例からも OSS の著作権を遵守することが必要であることがわかる．

特定の企業や団体，個人が占有するソフトウェアの場合，ソフトウェアの著作権の所在は明確である．OSS の場合であっても一部の組織は，OSS の厳格な所有権を維持している．たとえば，MySQL の著作権所有者である Oracle は，すべてのソースコードの作成者がプロジェクトにソースコードを追加するときに著作権を譲渡することを要求しており，Oracle が唯一の著作権所有者であることが保証されている [2]．しかし OSS においてこのような要件を定めることは稀なケースであり，多くの OSS プロジェクトの場合，著作権の譲渡は要求されておらず，ソースコードの作成者がそのソースコードの著作権を保持できる．そのため，多くの OSS においては著作権保持者が異なるソースコードが混在している．

また，ソースコードの作成者は増えていくため，ファイルによってはトークン単位で作成者が異なることも珍しくない．図 1 は Linux カーネル ver5.8 の crash_dump_64.c の一部である．ソースコードの各色はソースコードの作成者に対応しており，ソースコードの作成者が

```
 9  #include <linux/errno.h>
10  #include <linux/crash_dump.h>
11  #include <linux/uaccess.h>
12  #include <linux/io.h>
13
14  static ssize_t __copy_oldmem_page(unsigned long pfn, char *buf, size_t csize,
15                                    unsigned long offset, int userbuf,
16                                    bool encrypted)
17  {
18      void  *vaddr;
19
20      if (!csize)
21          return 0;
22
23      if (encrypted)
24          vaddr = (__force void *)ioremap_encrypted(pfn << PAGE_SHIFT, PAGE
25  _SIZE);
26      else
27          vaddr = (__force void *)ioremap_cache(pfn << PAGE_SHIFT, PAGE_SIZ
28  E);
29
30      if (!vaddr)
31          return -ENOMEM;
32
33      if (userbuf) {
34          if (copy_to_user((void __user *)buf, vaddr + offset, csize)) {
35              iounmap((void __iomem *)vaddr);
36              return -EFAULT;
```

図 1: 作成者ごとにトークンが色分けされているソースコード (深緑:Vivek Goyal, 青:Gustavo Padovan, 橙:Lianbo Jiang, 黄緑:Cliff Wickman, 茶:Akinobu Mita)

トークン単位で異なっている様子を示している．著作権を譲渡していなければ各ソースコードの作成者は独立に著作権を保持しているため，このようなソースコードではトークン単位で著作権の所在が異なる可能性がある．したがって，OSS における著作権の所在は複雑である．

OSS の著作権保持者を把握するにあたって有効な方法の 1 つがソースコードにコメントとして記載されている著作権表示を確認することである．著作権表示の一例を図 2 に示す．この著作権表示はソースコードの著作権が Google 社にあることを示している．ソースコードの著作権表示を自動抽出するため，The Linux Foundation の加盟企業である Hewlett Packard が主体となり FOSSology と呼ばれるオープンソースライセンスコンプライアンスソフトウェアシステムが開発されている [6]．ただし，ソースコードの作成者が必ずしも著作権表示を追加しているとは限らない．そのため，ソースコードのコメント内に記載されている著作権表示を検出するだけでは，一部の著作権保持者を見逃す可能性がある．

そこで著作権表示によってすべての著作権保持者が明確になるように著作権表示が正確に

```
/*
 * Greybus debugfs code
 *
 * Copyright 2014 Google Inc.
 *
 * Released under the GPLv2 only.
 */
```

図 2: ソースコードに記載された著作権表示の一例

なされることが求められる．

しかし，OSS と著作権の問題を扱った研究は前例が少なく，特に著作権表示の有無を扱った研究はまだない．たとえば Qiu らは著作権表示の内容と実際にソースコードを作成した団体や人物との差異について研究し [11]，Penta らは著作権表示の内容などからソースコードの作成者を追跡し，その人物がどのような変更を加えているかについて研究 [3] したが，OSS において著作権表示がどの程度あるかについては研究されていない．そこで，本研究では上記の研究の不足を補うために著作権表示の有無に関して研究する．

3　研究概要

本研究では OSS における著作権表示の実態を知るため，OSS の一例である Linux カーネルを研究対象として以下の 3 つの項目に注目して著作権表示があるかどうかを調査した．また，2.3 節で述べたように OSS のソースコードの著作権保持者はトークン単位で異なる可能性があるため，調査はトークン単位で行った．

- RQ1:Linux カーネルにおいて著作権表示があるソースコードの割合はどの程度か？
- RQ2:団体での開発と個人での開発によって著作権表示があるソースコードの割合に違いはあるのか？
- RQ3:機能ごとに分けられたディレクトリによって著作権表示があるソースコードの割合に違いはあるのか？

この章では調査にあたって使用したサービス・ツール・調査上の定義について説明した上で，調査方法について述べる．

3.1　調査に使用したツール

3.1.1　git

git とは無料の分散型バージョン管理システムである [1]．バージョン管理システムとはプロジェクトに対する過去の変更や状態を蓄積したものである．この蓄積されたものをリポジトリや git リポジトリと呼ぶ．また，変更をコミット (commit) と呼び，コミットはそれらを一意に識別するための commit ID を持つ．また，git には編集履歴を利用した git コマンドと呼ばれる様々なコマンドが用意されている．例えば，git コマンドを使えば commit ID からコミットの内容やコミットで追加したソースコードの作成者に関する情報などを取得することや，行単位でソースコードの作成者を調べることが可能となる．

3.1.2　GitHub

GitHub とは，git リポジトリをインターネット上に公開し世界中の人々と git のシステムで共同開発することを可能にしたプラットフォームである [5]．利用者は GitHub 上の任意の git リポジトリをローカルリポジトリにコピーすることで，ローカルリポジトリでコピーした git リポジトリを操作することが可能となる．世界中の人々が自分のソースコードを GitHub を通して保存・公開しており，多くの OSS も GitHub を利用することによってソースコードを公開している．現在 Linux カーネルの情報も GitHub 上に保存・公開されている．本調査では GitHub から Linux カーネルの git リポジトリを取得した．

表 1: debugfs.c に cregit を使用して調査用に抽出した情報

作成者名	追加したトークン	追加したコミットの commit ID(桁数を省略)
Greg Kroah-Hartman	42	27fb831, 48f7047, eb50fd3, ec0ad86
Alexandre Bailon	17	e8f824b
Viresh Kumar	4	f66832d
Alex Elder	1	47ed2c9, a46e967

3.1.3 cregit

cregit とは Daniel German らによって開発された，git のコミット情報を元にソースコードの作成者をトークン単位で調べることが可能なツールである [4]．このツールを用いることにより，Linux カーネルの各ファイルにおいて誰がどのトークンを追加したのか，追加したトークンはいくつかなどの情報を知ることができる．表 1 は，例として Linux カーネルのファイル debugfs.c に cregit を使用して本調査に必要となる debugfs.c 中のソースコードの作成者，それぞれが追加したトークン数，それぞれが追加したコミットの commit ID の情報を抽出した結果を示している．

3.2 調査における定義

3.2.1 著作権表示があるトークンの定義

著作権表示をファイルに追加した団体や人と同じ団体や人が同じファイル内に追加したトークンの著作権保持者はその著作権表示によって示される．本研究ではこのようなトークンを著作権表示があるトークンと定義する．図 3 は著作権表示があるトークンの一例である．25 行目の “struct” というトークンを追加した Kevin Hilman は著作権表示をファイルに追加している．この時，“struct” というトークンの著作権の所在は著作権表示によって示されている．本研究ではこのようなトークンを著作権表示があるトークンとして扱う．

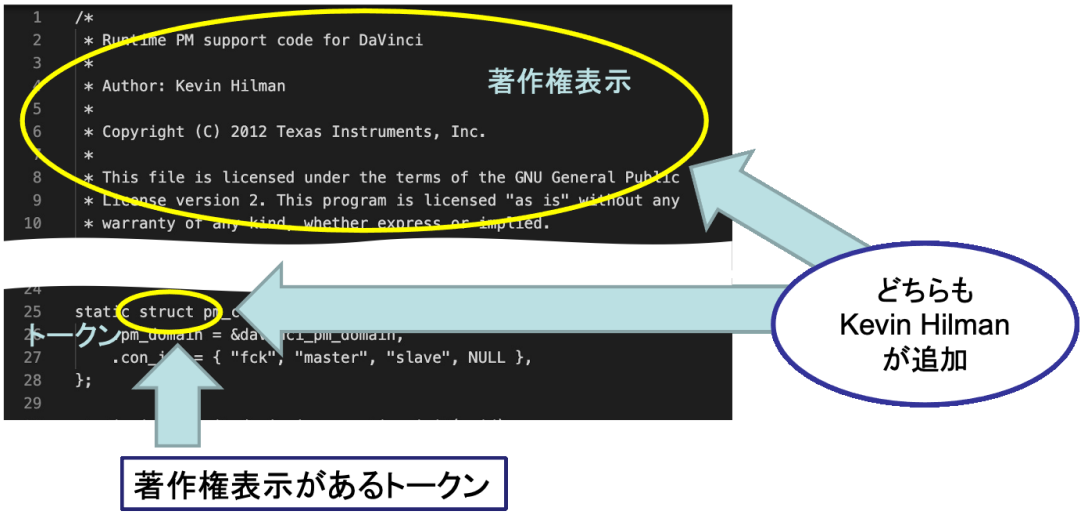

図 3: 本研究において著作権表示があるトークンと定義されるトークンの例

3.2.2 ソースコードの作成者が団体に所属していることの定義

本研究では職務著作の考えに基づくためソースコードの作成者が団体に所属していること，所属していないことを定義しておく必要がある．Qiu らの論文ではメールアドレスのドメインを用いてソースコードの作成者が所属する団体を判断している [11]．それにならい，本調査においてはメールアドレスのドメインを元に団体に所属していることを定義する．git の開発においてはソースコードの作成者はコミットを追加する際にメールアドレスの登録が義務付けられており，そのメールアドレスは git コマンドによって確認することが可能である．また，一般にメールサービスには所属する団体から支給される独自ドメインメールと個人的に取得することが可能なフリーメールが存在し，両者はメールアドレスのドメインを確認することによって判別することが可能である．よって本調査では個人的に取得可能なフリーメールのドメインのメールアドレスを使用しているソースコードの作成者を個人開発者と定義し，それ以外のドメインのメールアドレスを使用しているソースコードの作成者を団体に所属していると定義した．ドメインがフリーメールのドメインであることの判断基準は T. Brian Jones によって現時点で 3,782 のフリーメールのドメインがまとめられたリストに含まれていることとした[1]．また，本調査においてはソースコードの作成者同士が同じ団体に所属しているかどうかもメールアドレスのドメインを用いて定義する．例えば，A 氏のメールアドレスのドメインと B 氏のメールアドレスのドメインが同じであったとする．このとき，そのドメインがフリーメールのドメインでないならば A 氏と B 氏は団体に所属していると定義される．また，その上で両者のメールアドレスのドメインが同じであることから A

[1]free_email_provider_domains https://gist.github.com/tbrianjones/5992856/

氏と B 氏は同じ団体に所属していると定義される．

3.3 調査方法

この節ではまず，本研究の調査対象と調査に共通するトークンと著作権表示の分析方法について述べる．その後，調査手順とその詳細について述べる．

3.3.1 調査対象

本調査では調査開始時 (2020 年 8 月) の最新バージョンであった Linux カーネル ver5.8 を調査対象とした．本調査では Linux カーネルの情報を GitHub から取得している．しかし，GitHub で現行の Linux が管理され始めたのは ver2.6.12_rc2 からである．そのため，それ以前から存在するソースコードのすべては ver2.6.12_rc2 を GitHub で公開した Linus Torvalds が作成した扱いになる．よって ver2.6.12_rc2 以前から存在するファイルに対して git コマンドを実行しても正しい編集履歴が返されるとは限らない．したがって今回の調査では ver5.8 に存在するファイルのうちファイル名の変更によって特定しきれなかったもの以外の ver2.6.12_rc2 から存在するファイルを除外したものを調査対象とした．

3.3.2 トークンと著作権表示の分析方法

2.3 節で述べたようにソースコードの著作権保持者はトークン単位で異なる可能性がある．そのためソースコードの作成者を git コマンドを用いて行単位で調査した場合，トークンを追加・変更した人物は変更したトークンを含む行すべての作成者と示され，もともとその行を開発していた人物の貢献は失われてしまう．例えば，図 4 と図 5 はそれぞれ Linux カーネル ver5.8 のファイル debugfs.c に対する git コマンドを用いた行単位の調査と cregit を用いたトークン単位の調査の結果の一部である．図 4 のソースコードの色はソースコードの作成者の色に対応している．トークン単位の調査結果である図 4 の 14 行目を見ると AlexElder は “__init” しか追加していないことがわかるが，行単位の調査結果である図 4 の 14 行目を見ると Alex Elder は 14 行目の作成者として示されている．このように，行単位の調査ではソースコードに対する作成者の特定精度がトークン単位の調査に比べて低くなる．

よって本調査では cregit を用いてトークン単位で著作権表示の有無を調査する．しかし，cregit を用いたトークン単位の調査ではコメントを 1 つのトークンとして扱うため，コメントの作成者は 1 人のみしか示されない．例えば，前述した図 4 と図 5 において行単位の調査結果である図 4 の 2 行目から 7 行目を見ると Greg Kroah-Hartman と Alex Elder が著作権表示を追加していることがわかるが，トークン単位の調査結果である図 4 の 2 行目から 7 行目を見ると Greg Kroah-Hartman のみが著作権表示の作成者として示されている．よって

Contributors: 4

Author	Tokens	Token Proportion	Commits
Greg Kroah-Hartman	42	65.62%	6
Alexandre Bailon	17	26.56%	1
Viresh Kumar	4	6.25%	1
Alex Elder	1	1.56%	1
Total	64		9

```
1   // SPDX-License-Identifier: GPL-2.0
2   /*
3    * Greybus debugfs code
4    *
5    * Copyright 2014 Google Inc.
6    * Copyright 2014 Linaro Ltd.
7    */
8
9   #include <linux/debugfs.h>
10  #include <linux/greybus.h>
11
12  static struct dentry *gb_debug_root;
13
14  void __init gb_debugfs_init(void)
15  {
16      gb_debug_root = debugfs_create_dir("greybus", NULL);
17  }
18
```

図 4: 作成者ごとにトークンが色分けされた Linux カーネル ver5.8 のファイル debugfs.c に対するトークン単位の調査結果の一部

トークン単位の調査ではコメントの作成者の特定精度のみ行単位の調査に比べて低くなる．そこでコメントに記載される著作権表示の作成者の特定のみ git コマンドを用いて行単位で調査する．

したがって，本研究では基本的に cregit を用いてトークン単位で調査するが，著作権表示を追加した人物を出来る限り正確に特定するために著作権表示のみ行単位で調査する．

3.3.3 調査手順

本研究における調査手順について述べる．まず，Linux カーネルを元にすべての RQ に共通のデータセットを作成し，各 RQ に沿って著作権表示があるトークンをカウントする．この節以降では，すべての RQ に共通するデータセットの作成方法と著作権表示があるトークンのカウント方法について述べる．

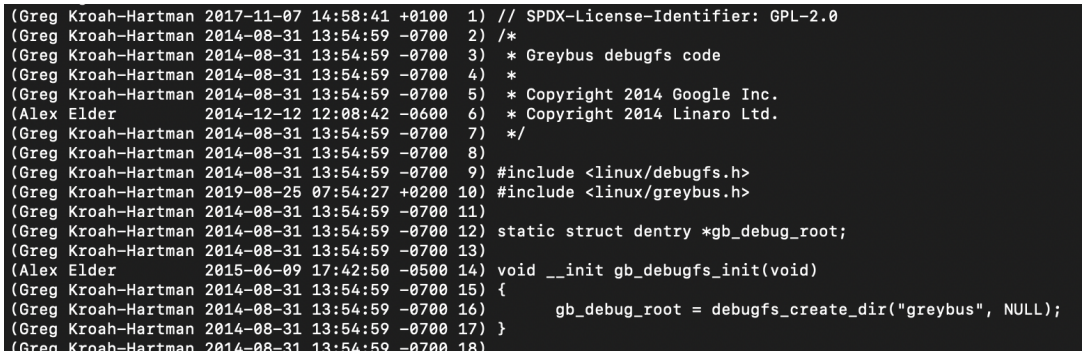

図 5: Linux カーネル ver5.8 のファイル debugfs.c に対する行単位調査の結果の一部

3.3.4 データセットの作成方法

調査を行うにあたってソースコードの作成者に関する情報をまとめたデータセットと，著作権表示を追加したコミットに関する情報をまとめたデータセットを作成する．この節ではそれらのデータセットの作成方法について述べる．

はじめに前処理として，調査対象ファイルをリスト化する．まず，GitHub から Linux カーネルの git リポジトリをクローンする．そのリポジトリから git コマンドを用いて ver5.8 の状態と ver2.6.12_rc2 の状態を作る．そして ver5.8 に存在するファイルのパスと ver2.6.12_rc2 に存在するファイルのパスを比較し両方に存在するファイルを除く．さらに ver5.8 に存在するファイルのコミット履歴を辿り ver2.6.12_rc2 のファイルを追加したコミットが確認できた場合はそのファイルも除く，そして残ったファイルを調査対象としてリスト化した．

次にソースコードの作成者情報のデータセットを作成する．図 6 はソースコードの作成者情報のデータセットを作成する流れを示している．Linux カーネルの git リポジトリ内の調査対象ファイルに対して cregit を使用し，各ファイルに対してソースコードの作成者とそれぞれが追加したコミットの ID，トークンの総数を取得する．そして git コマンドを用いることでソースコードの作成者のメールアドレスのドメインを取得した．

最後に著作権表示を追加したコミット情報のデータセットを作成する方法について述べる．図 7 は著作権表示を追加したコミット情報のデータセットを作成する流れを示している．Linux カーネルの git リポジトリ内の調査対象ファイルに対して，git コマンドを用いることでソースコードを行単位で分析し，著作権表示を追加したコミットの commit ID とそのコミットに使用されたメールアドレスのドメインを取得した．

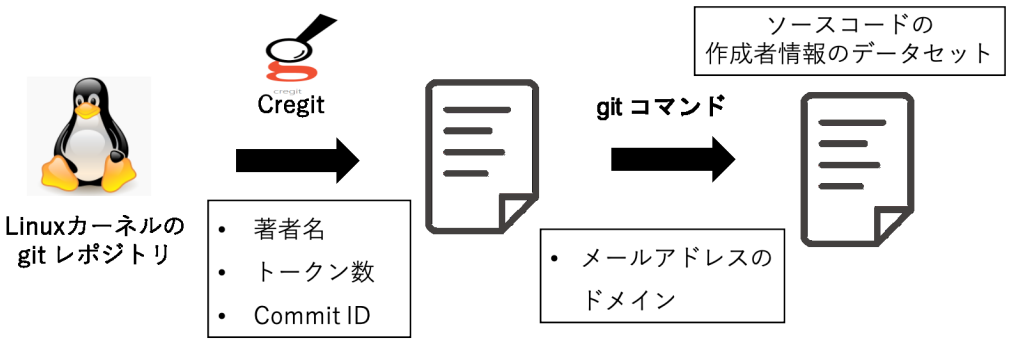

図 6: ソースコードの作成者情報のデータセットを作成する流れ

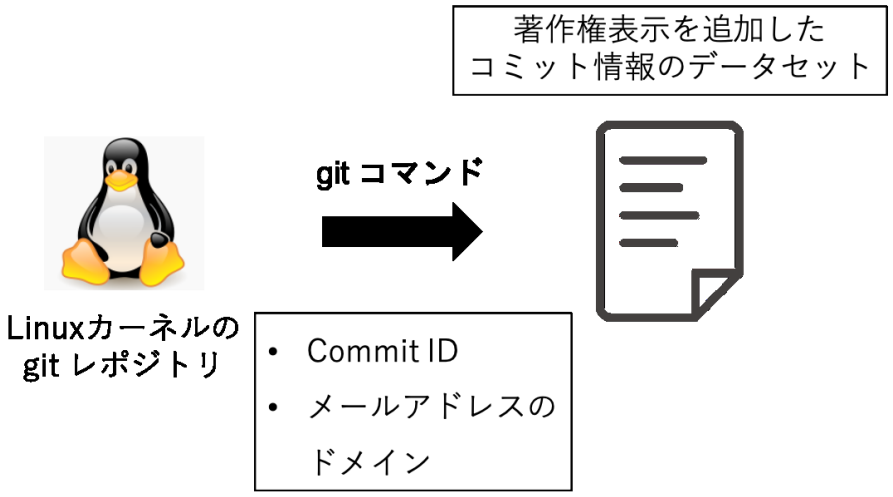

図 7: 著作権表示を追加したコミット情報のデータセットを作成する流れ

3.3.5 著作権表示があるトークンのカウント方法

調査対象の各ファイルに対してソースコードの作成者情報のデータセットと，著作権表示を追加したコミット情報のデータセットの情報をもとに著作権表示があるトークンの数をカウントする．著作権表示があるトークンの判断基準は 3.2.1 節で述べたとおりである．例として Linux カーネルのファイル debugfs.c に対するデータセットと集計結果を載せる．表 2 は Linux カーネルのファイル debugfs.c に対するソースコードの作成者情報のデータセットであり，表 3 は Linux カーネルのファイル debugfs.c に対する著作権表示を追加したコミット情報のデータセットである．この場合，表 3 より著作権表示を追加したコミットは 2 つであり，それらのコミットに使用されたメールアドレスのドメインは kroah.com と linaro.org である．この 2 つのドメインはどちらも個人のドメインではないため，これらのドメイン

表 2: debugfs.c に対するソースコードの作成者情報のデータセット

作成者名	追加したトークン	追加したコミットの commit ID(桁省略)	メールアドレスの ドメイン
Greg Kroah-Hartman	42	27fb831, 48f7047, eb50fd3, ec0ad86	kroah.com, linuxfoundation.org
Alexandre Bailon	17	e8f824b	baylibre.com
Viresh Kumar	4	f66832d	linaro.org
Alex Elder	1	47ed2c9, a46e967	linaro.org

表 3: debugfs.c に対する著作権表示を追加したコミット情報のデータセット

著作権表示を追加したコミットの commit ID(桁省略)	メールアドレスのドメイン
27fb831	kroah.com
a46e967	linaro.org

を使用する団体の著作権表示が書かれたものと判断する．よって kroah.com を使ってソースコードを作成した Greg Kroah-Hartman と，linaro.org を使ってソースコードを作成した Viresh Kumar と Alex Elder が追加したトークンの合計を著作権表示があるトークンとしてカウントする．したがって集計結果は表 4 のようになる．

表 4: debugfs.c の集計結果

	総数
著作権表示があるトークン	47
著作権表示がないトークン	17

表 5: Linux カーネル ver5.8 の著作権表示があるトークンの割合の調査結果

	総数	全トークンに占める割合
著作権表示があるトークン	70,314,586	77.54%
著作権表示がないトークン	20,370,929	22.46%

4 Linux カーネルにおける著作権表示の調査

作成したデータセットを元に著作権表示の有無に関して調査した．この章では各 RQ の調査結果と調査結果に対する考察について述べる．

4.1 RQ1:Linux カーネルにおいて著作権表示があるソースコードの割合はどの程度か？

RQ1 では，まず Linux カーネルにおける著作権表示があるトークンの割合と著作権表示がないトークンを含まないファイル，すなわちファイル中のすべてのトークンに著作権表示があるファイルの割合を調査する．そしてそれらの割合が過去と比較してどのように変化しているかを調査した．

著作権表示があるトークンの割合の調査

はじめに，Linux カーネル全体に対して著作権表示があるトークンの割合を調査した．表 5 は Linux カーネル ver5.8 の著作権表示があるトークンの割合の調査結果を示したものである．調査対象となった全トークンのうち著作権表示があるトークンは 77.54%であり，著作権表示がないトークンは 22.46%であった．

すべてのトークンに著作権表示があるファイルの割合の調査

次に，すべてのトークンに著作権表示があるファイルの割合を調査した．著作権表示があるトークンの割合の調査と同様にトークンごとの著作権表示の有無を判定した上で，ファイルごとに集計を行い，そのファイル内のすべてのトークンに著作権表示があるファイルの数をカウントした．また，ファイル内に著作権表示を 1 つ以上含むファイルの割合に関しても調査を行った．表 6 は Linux カーネル ver5.8 のすべてのトークンに著作権表示があるファイルの割合に関する調査結果を示したものである．調査対象となった全ファイルのうち著作権表示を含むファイルは 81.03%，著作権表示を含まないファイルは 18.97%であり，またファイル中のすべてのトークンに著作権表示があるファイルは 15.73%であった．

表 6: Linux カーネル ver5.8 のすべてのトークンに著作権表示があるファイルの割合に関する調査結果

	総数	全ファイルに占める割合
著作権表示を含むファイル	34,599	81.03%
著作権表示を含まないファイル	8,102	18.97%
すべてのトークンに著作権表示があるファイル	6,719	15.73%

表 7: inux カーネル ver5.8 と Linux カーネル ver4.19 の著作権表示があるトークンに関する割合の比較

	ver4.19 全体	ver5.8 全体
著作権表示があるトークン	116,366,185	70,314,586
著作権表示がないトークン	16,759,150	20,370,929
著作権表示を含むファイル	30,411	34,599
著作権表示を含まないファイル	6,862	8,102
すべてのトークンに著作権表示があるファイル	7,738	6,719

時間による割合の変化の調査

最後に，時間による割合の変化を調査した．ここでは ver5.8 以前のバージョンと比較するため ver5.8 の 2 年前にリリースされた ver4.19 を対象に前述した ver5.8 に対する調査方法と同じ方法で著作権表示があるトークンの割合とすべてのトークンに著作権表示があるファイルの割合に関する調査を行い，その結果を比較した．表 7 に調査結果を示す．ソフトウェアの開発が進むにつれ，著作権表示があるトークンは減少し，著作権表示がないトークンが増加していることがわかる．著作権表示を含むファイルの数は増加しているが，すべてのトークンに著作権表示があるファイルの数は減少している．この差は統計的に有意なものであった．また、図 8 と図 9 はそれぞれ Linux カーネル ver5.8 と Linux カーネル ver4.19 における著作権表示があるトークンの割合と，すべてのトークンに著作権表示があるファイルの割合の比較を示したものである．割合で見ても，著作権表示があるトークンの割合と，すべてのトークンに著作権表示があるファイルの割合が減少していることがわかる．

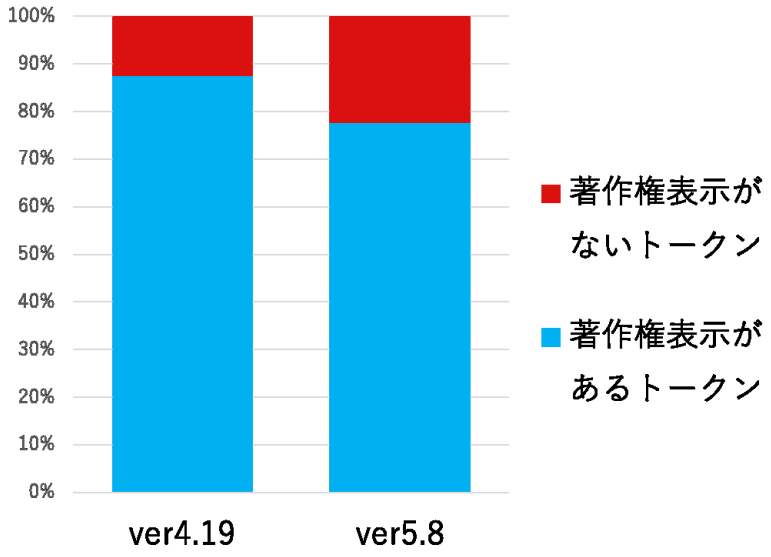

図 8: Linux カーネル ver5.8 と Linux カーネル ver4.19 の著作権表示があるトークンの割合の比較

RQ1 の考察

著作権表示があるソースコードの割合，著作権表示を含むファイルの割合はともに 80% 前後と高いように見える．しかし，すべてのトークンの著作権の所在が明らかであることが望まれるなかで約 20%ものトークンの著作権の所在が不明であるというのは好ましい状況とは言えない．また，ファイル内のすべてのトークンに著作権表示があるファイルの割合は 15.7%と非常に低く，ファイルに含まれる著作権表示がそのファイル内のすべてのトークンの著作権の所在を明らかにできていない場合が多いことがわかる．さらに，過去バージョンとの比較においては著作権表示を含むファイルの数は増えているにも関わらず，著作権表示があるトークンと，ファイル内のすべてのトークンに著作権表示があるファイルの数と割合は減少している．この結果から，著作権の所在を著作権表示だけで確認することが難しくなっていることがわかる．

このような結果となった原因として，ソースコードの作成者が著作権表示を追加することを忘れていた，意図的に行わなかったということが考えられる．また，著作権があるトークンの総数が減っていることからリファクタリングやソースコードの修正があったことがうかがえる．そこで，リファクタリングの際に著作権表示を追加することを忘れていた，修正箇所が多くないために著作権表示の追加を遠慮したというようなことがあったのではないかと推測される．

そこで著作権表示が適切に追加されるように，ソースコードが変更を加えられる際，変更

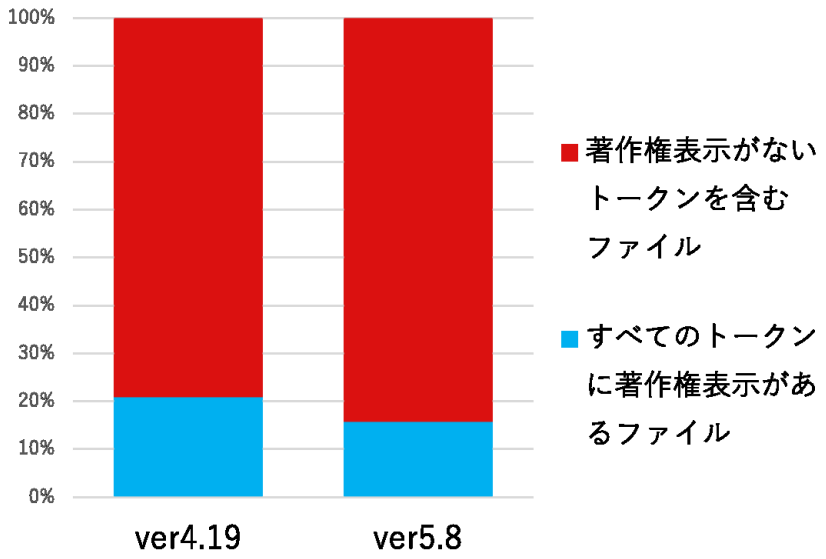

図 9: Linux カーネル ver5.8 と Linux カーネル ver4.19 のすべてのトークンに著作権表示があるファイルの割合の比較

によって著作権表示がないトークンが新たに発生する場合は警告するシステムの作成が望まれる．このシステムを git に導入することができれば，著作権表示の追加し忘れは未然に防がれ，著作権表示を追加することに対する遠慮も薄まると考えられる．

4.2 RQ2:団体での開発と個人での開発によって著作権表示があるソースコードの割合に違いはあるのか？

RQ2 ではソースコードが団体によって開発されている場合と，個人で開発されている場合においてどのような違いがあるかを調査した．まず，それぞれの場合において追加されたすべてのトークンに対する著作権表示の有無について調査する．次に，各団体・各個人開発者に焦点を当てそれぞれが追加したトークンのうち著作権表示があるトークンの割合を調査した．

団体での開発と個人での開発による著作権表示があるソースコードの割合の調査

はじめに，団体での開発によって追加されたトークンと個人での開発によって追加されたトークンにおける著作権表示の有無について調査し，その結果を比較した．表 8 と表 9 はそれぞれの場合における著作権表示があるトークンに関する調査結果の総数と割合を示したものである．団体での開発によって追加されたトークンのうち著作権表示があるトークンの

表 8: 団体での開発における著作権表示があるトークンに関する調査結果の総数と割合

	総数	割合
著作権表示ありトークン	66,977,741	78.96%
著作権表示なしトークン	17,849,760	21.04%

表 9: 個人での開発における著作権表示があるトークンに関する調査結果の総数と割合

	総数	割合
著作権表示ありトークン	6,018,202	70.48%
著作権表示なしトークン	2,521,169	29.52%

割合は 78.96%であり，個人での開発によって追加されたトークンのうち著作権表示があるトークンの割合は 70.48%であった．

団体での開発における著作権表示があるトークンの割合の調査

次に，各団体別に追加したトークンのうち著作権表示があるトークンの割合を調査した．図 10 は追加したトークンのうち著作権表示があるトークンの割合と団体の分布を示したものである．著作権表示があるトークンの 10%以下の団体が最も多く，次に著作権表示があるトークンの割合が 90%より大きい団体が多いことがわかる．

個人での開発における著作権表示があるトークンの割合の調査

最後に，各個人開発者別に追加したトークンのうち著作権表示があるトークンの割合を調査した．図 11 は追加したトークンのうち著作権表示があるトークンの割合と個人開発者の分布を示したものである．著作権表示があるトークンの 10%以下の個人開発者が最も多く，次に著作権表示があるトークンの割合が 90%より大きい個人開発者が多いことがわかる．

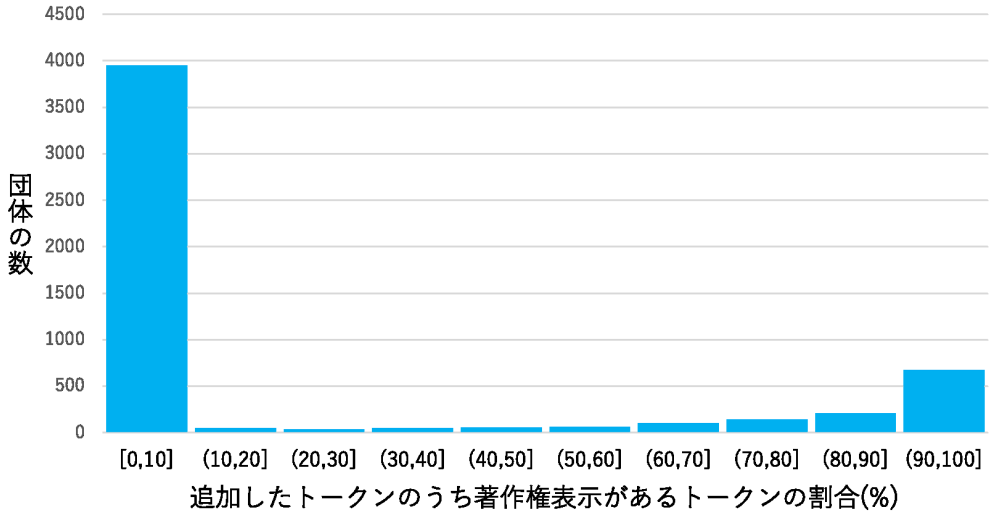

図 10: 追加したトークンのうち著作権表示があるトークンの割合と団体の分布

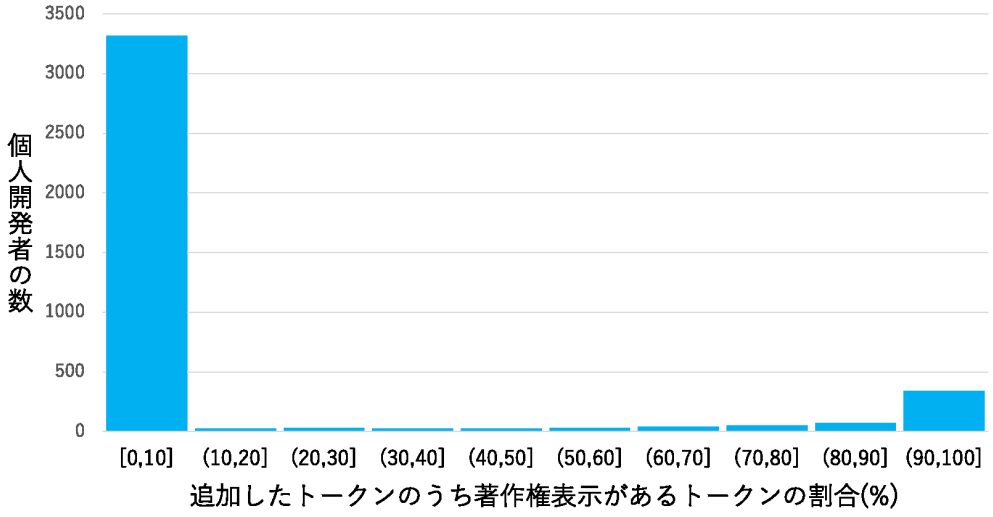

図 11: 追加したトークンのうち著作権表示があるトークンの割合と個人開発者の分布

RQ2 の考察

著作権表示があるトークンの割合は団体での開発でも個人での開発でも 70%から 80%の間であり，団体での開発の方がやや著作権表示があるトークンの割合が高かったものの，大きな違いは見られなかった．

また，団体別に著作権があるトークンの割合について調べたところでは著作権表示があるトークンの割合が非常に低い団体の数が最も多く，次に著作権表示があるトークンの割合が高い団体の数が多かった．さらに，個人開発者別に著作権があるトークンの割合について調べたところでは著作権表示があるトークンの割合が低い作成者の数が最も多かった．よって団体別の分布も個人開発者別の分布もよく似ていた．

したがって調査結果では著作権表示の有無に関して団体での開発と個人での開発の間に大きな違いは見られなかった．

団体での開発の方がやや著作権表示があるトークンの割合が高かった原因としては団体での開発の方が著作権表示を管理している割合が高いことが推測される．また，団体での開発か個人での開発かによる差が大きくは出なかったのは団体のドメインを使っていても実際には個人で開発している人や個人のドメインを使っていても実際には団体で開発している人が少なからずいた可能性や，元から著作権表示の有無が団体での開発か個人での開発かに影響を受ける性質のものではない可能性が考えられる．

また，著作権表示があるトークンの割合は団体での開発でも個人での開発でも 70%以上あったにも関わらず，追加したトークンのうち著作権表示があるトークンの割合が低い団体や個人開発者が非常に多いことから．著作権表示があるトークンの割合が低い団体や個人開発者が追加したトークン数は少ないのではないかと考え，追加で調査を行なった．図 12 は追加したトークンのうち著作権があるトークンの割合が 10%以下だった団体における追加したトークン数の内訳を示したものである．また，図 13 は追加したトークンのうち著作権があるトークンの割合が 10%以下だった個人開発者における追加したトークン数の内訳を示したものである．どちらの結果においても追加したトークン数が少ない団体と人が大半を占めていることがわかる．よって，この 2 つの図から著作権表示を追加しない一因として追加しているトークンの数が少ないことがあげられる．

このような結果となった原因として，著作権表示を追加することに対して追加したトークンが少ないことからくる遠慮があったのではないかと考えられる．したがって，著作権表示があるトークンの割合を高めるには 4.1 節に述べたシステムを導入することが有効だと考えられる．

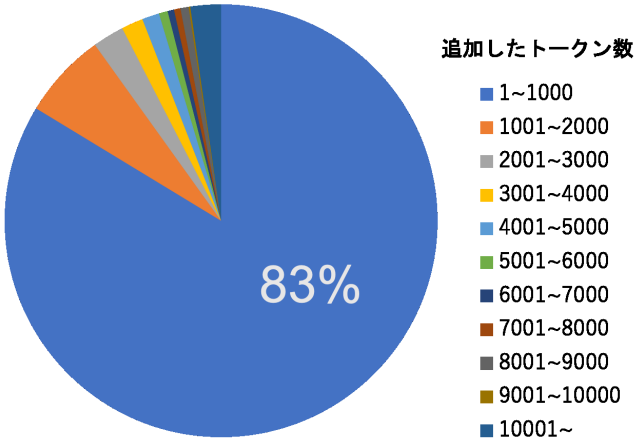

図 12: 追加したトークンのうち著作権があるトークンの割合が 10%以下だった団体における追加したトークン数の内訳

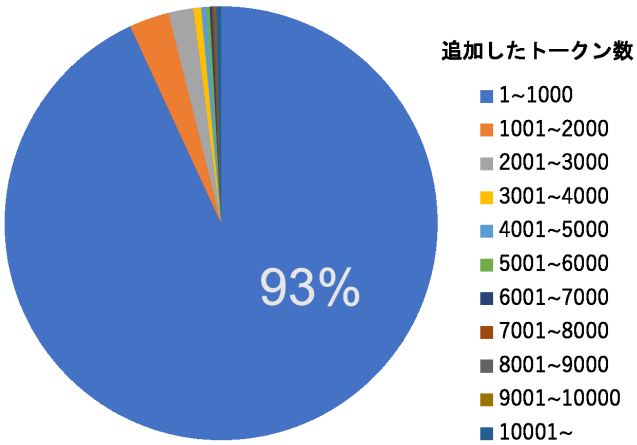

図 13: 追加したトークンのうち著作権があるトークンの割合が 10%以下だった個人開発者における追加したトークン数の内訳

表 10: それぞれのディレクトリにおける調査対象となったファイルとトークンの総数

ディレクトリ名	ファイル総数	トークン総数
Documentation	2	789
arch	7,481	6,071,290
block	66	175,061
certs	5	1,730
crypto	130	348,136
drivers	23,033	66,316,767
fs	1,125	3,421,135
include	4,446	2,108,860
init	2	1,191
ipc	4	3,652
kernel	362	1,046,706
lib	330	712,445
mm	91	287,626
net	1,222	3,094,129
samples	177	153,933
scripts	61	114,297
security	143	240,589
sound	1,647	3,968,254
tools	2,364	2,588,652
virt	10	30,273

4.3 RQ3:機能ごとに分けられたディレクトリによってよって著作権表示があるソースコードの割合に違いはあるのか？

RQ3 では Linux カーネルの中で機能ごとに分けられているディレクトリにおける著作権表示があるトークンの割合を個別に調査し，その結果を比較した．表 10 は調査対象となったファイルとトークンの総数，表 11 はそれぞれのディレクトリにおける著作権表示があるトークンに関する割合を示したものである．各ファイルごとに対象となったファイル数やトークン数は大きく異なる．また，著作権表示に関する割合も大きく異なっていることがわかる．

表 11: それぞれのディレクトリにおける著作権表示があるトークンに関する割合

ディレクトリ名	著作権表示があるトークンの割合	著作権表示を含むファイルの割合	すべてのトークンに著作権表示があるファイルの割合
Documentation	0.00%	0.00%	0.00%
arch	61.00%	68.44%	6.99%
block	47.33%	54.55%	4.55%
certs	79.08%	40.00%	0.00%
crypto	71.59%	92.31%	4.62%
drivers	83.23%	92.86%	21.66%
fs	65.98%	85.33%	4.44%
include	56.57%	60.95%	12.66%
init	9.66%	50.00%	0.00%
ipc	27.19%	75.00%	0.00%
kernel	49.21%	65.75%	3.87%
lib	60.33%	57.27%	16.67%
mm	42.54%	62.64%	0.00%
net	60.31%	78.81%	3.52%
samples	72.38%	68.36%	5.08%
scripts	64.46%	80.33%	8.20%
security	79.17%	93.01%	4.90%
sound	80.59%	90.77%	9.35%
tools	42.97%	42.51%	12.61%
virt	33.86%	90.00%	0.00%

表 12: 機能別の著作権表示に関する割合との相関係数

	ファイル総数	1 ファイルあたりのトークン数	1 ファイルあたりのコミット数	1 ファイルあたりの作成者数
著作権表示があるトークンの割合	0.331	0.246	-0.099	-0.078
著作権表示を含むファイルの割合	0.250	0.594	0.271	0.350
すべてのトークンに著作権表示があるファイルの割合	0.688	0.127	-0.410	-0.420

RQ3 の考察

ディレクトリによって調査対象となったファイル数が大きく異なるために割合だけに注目して比較することは適切ではない可能性があるが，結果だけに言及すれば各割合はディレクトリごとに大きく異なっている．この違いを生んだ原因を探るため，それぞれのディレクトリごとにファイル総数，1 ファイルあたりのコミット数，1 ファイルあたりのソースコードの作成者数，1 ファイルあたりのトークン数と著作権表示に関する割合との相関係数を調査した．表 12 はその結果を示している．ここでコミット数，ソースコードの作成者数，トークン数を 1 ファイルで求めているのは総数にするとファイル数による影響が大きいと考えたからである．相関係数からファイル総数とすべてのトークンに著作権表示があるファイルの割合の間には正の相関があると言える．すべてのトークンに著作権表示があるファイルには著作権表示が管理されたファイル，または作成時以降編集されていないファイル，1 社または 1 個人によってのみ管理されているファイルが存在すると考えられる．よってこの結果からファイル総数が多いディレクトリではリファクタリングなどが十分に進んでおらず未編集のファイルが多く残されているのではないかと推測した．また，その他の関係については相関が高いとは言えず，著作権表示の有無に関する割合に違いが出る詳しい原因を突き止めることはできなかった．したがって，これらの違いが生まれた背景には確かな原因が存在しないか，今回調べるまでには至らなかった別の原因が潜んでいるのではないかと考えられる．例えば，ソースコードの再利用数の違いや著作権表示を付与することに関する規則がコメントや専用のファイルなどによって定められているかどうかの違いなどが考えられる．

5　妥当性への脅威

この章では調査方法の性質上，集計に誤りが生じると考えられる場合とその原因について述べる．

5.1　著作権表示があるトークン数のカウントに誤りが生じる場合

本研究では職務著作の考えに基づいて調査しており，ソースコードの作成者が団体に所属しているかどうかは作成者が開発時に使用していたメールアドレスのドメインで判断している．例えば，amd.com というドメインを使っている人は amd 社の業務としてソースコードを開発していると判断する．そして，この判断基準によって団体に所属していると判断された人物が著作権表示を追加した場合には，所属する団体の著作権表示が追加されたものだと判断し，同じファイルの中で同じドメインを使用して開発をおこなっている人物が追加したすべてのトークンを著作権表示があるトークンとしてカウントしている．したがってソースコードの作成者が個人のメールアドレスのドメインを使いながら団体の一員として開発している場合や，団体のドメインを使いながら個人的に開発している場合は著作権表示があるトークン数をカウントする際に誤りが生じてしまう．これは本研究の根幹に関わる問題あるため，どの程度そのような場合があるのかを知る必要がある．しかし，そのような割合を知るためには実際に各ソースコードの作成者にどのような状況で開発していたかを尋ねる他に方法がない．よって本研究ではこのような場合の発生数やその割合については確認できていない．

5.2　団体で開発した著作権表示があるトークン数のカウントに誤りが生じる場合

4.2 節では団体ごとに著作権表示があるトークン数と追加したトークン数の総数をカウントし，著作権表示があるトークンが占める割合を計算している．この調査においても，5.1 節で述べたようにソースコードの作成者が個人のメールアドレスのドメインを使いながら団体の一員として開発している場合や，団体のドメインを使いながら個人的に開発している場合にはトークン数のカウントに誤りが生じてしまう．さらに，この調査においてはソースコードの作成者が複数のメールアドレスを使用しており，かつそれらのドメインが異なるといった場合にも誤りが生じる可能性がある．例えば，あるファイルにおいて A というソースコードの作成者が 50 個のトークンを追加しており amd.com と ti.com という 2 つの団体から支給されるドメインのメールアドレスを使用していた場合，いくつのトークンが amd.com のメールアドレスで追加されており，いくつのトークンが ti.com のメールアドレスで追加されているかを知ることは今回の調査ではできなかった．そこでこの調査では著作権表示がある方を優先してカウントした．例えば，A のケースにおいて amd.com を使用する人物に

よって著作権表示が追加された場合には amd.com に 50 個の著作権表示があるトークンがあると判断し，ti.com を使用する人物によって著作権表示が追加された場合には ti.com に 50 個の著作権表示があるトークンがあると判断する．ここで問題になるのはどちらの著作権表示も追加されていた場合である．このような場合にはカウントの重複が発生してしまう．A のケースにおいては amd.com にも ti.com にも 50 個の著作権表示があるトークンがあると判断されてしまう．このような形で重複してカウントされたトークンは 2,681,357 個あり，団体で開発したと判断された全トークンのうち 3 %ほどであった．また，どちらの著作権表示も追加されていない場合にもどちらのトークンとして扱うかという問題が発生する．このような場合にはどちらにも含めず独立で扱うこととした．例えば，A のケースにおいて amd.com を使用する人物によっても ti.com を使用する人物によっても著作権表示が追加されなかった場合には，amd.com と ti.com というドメインを使う団体が 50 個の著作権表示がないトークンを追加したと扱った．このように複数ドメインの団体として扱いを受けたトークンは 3,204,251 個あり，団体で開発したと判断された全トークンのうち 3.8 %ほどであった．

5.3 団体数のカウントに誤りが生じる場合

5.2 節で述べたようにソースコードの作成者が複数のメールアドレスを使用していて，かつそれらのドメインが異なっており，さらにいずれのドメインを使用した人物からの著作権表示も追加されていない場合には，複数のドメインの組み合わせが 1 つの団体として独立に扱われる．そしてそのようなドメインの組み合わせに含まれるドメインが既に存在するドメインであった場合は団体数のカウントに誤りが生じる．このようなカウントの誤りは最大で 1135 件発生しており，団体と判断したドメインのうち 21 %にも及んだ．この結果は 4.2 節で述べた結果に影響を与えるものである．しかし，この 1,135 件を差し引いたとしてもグラフの様子は大きくは変わらない．図 14 は図 10 に示した，追加したトークンのうち著作権表示があるトークンの割合と団体の分布の調査結果からカウントが誤っている可能性があるものを除いて示したものである．この図からも著作権表示があるトークンの割合が非常に低い団体の数が最も多いという結果は変わらないことがわかる．

5.4 個人で開発した著作権表示があるトークン数のカウントに誤りが生じる場合

4.2 節では個人開発者ごとに著作権表示があるトークン数と追加したトークン数の総数をカウントし，著作権表示があるトークンが占める割合を計算している．この調査においても，5.1 節で述べたようにソースコードの作成者が個人のメールアドレスのドメインを使いながら団体の一員として開発している場合や，団体のドメインを使いながら個人的に開発してい

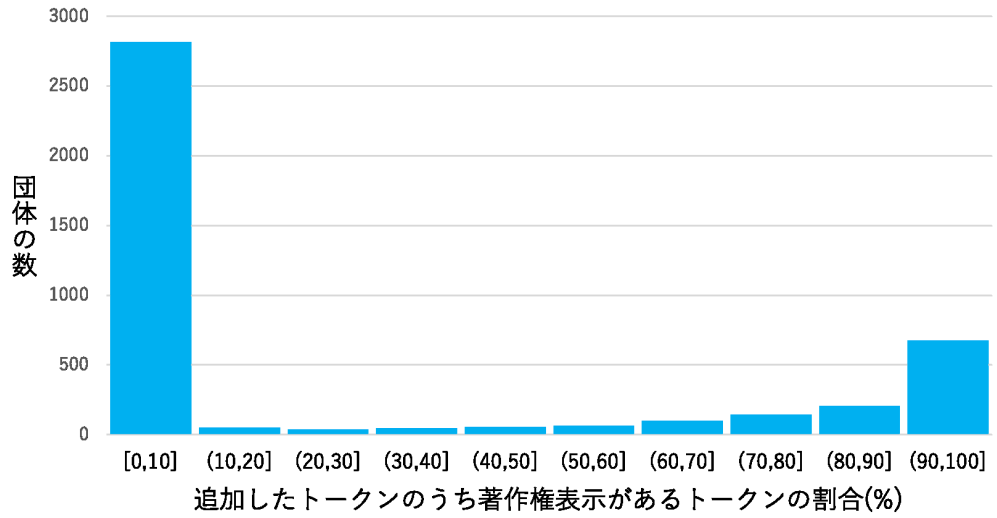

図 14: 図 10 に示した調査結果からカウントが誤っている可能性があるものを除いたグラフ

る場合にはトークン数のカウントに誤りが生じてしまう．さらに，この調査においては 5.2 節で述べた問題と同様の問題が発生する．つまり，ソースコードの作成者が複数のメールアドレスを使用しており，かつそれらのドメインが異なるといった場合にも誤りが生じる可能性がある．例えば，あるファイルにおいて B というソースコードの作成者が 50 個のトークンを追加しており amd.com と gmail.com という団体から支給されるドメインのメールアドレスと個人で取得できるドメインのメールアドレスを使用していた場合，いくつのトークンが amd.com のメールアドレスで追加されており，いくつのトークンが gmail.com のメールアドレスで追加されているかを知ることは本調査ではできなかった．そこでこの調査では著作権表示がある方を優先してカウントした．例えば，B のケースにおいて amd.com を使用する人物によって著作権表示が追加された場合には amd.com に 50 個の著作権表示があるトークンがあると判断し，gmail.com を使用して B が著作権表示を追加していた場合には個人開発者 B に 50 個の著作権表示があるトークンがあると判断する．ここで問題になるのはどちらの著作権表示も追加されていた場合である．このような場合にはカウントの重複が発生してしまう．B のケースにおいては amd.com にも個人開発者 B にも 50 個の著作権表示があるトークンがあると判断されてしまう．このような形で重複してカウントされたトークンは 490,606 個あり，団体で開発したと判断された全トークンのうち 0.5 %ほどであり，個人で開発したと判断された全トークンのうち 6 %ほどであった．そしてこのカウントの重複は 4.2 節の調査や 4.2 節の調査においても発生している．また，どちらの著作権表示も追加されていない場合にもどちらのトークンとして扱うかという問題が発生する．このような場合には 5.2 節で述べたように 1 つの団体として独立で扱うこととした．このような扱いとなっ

たトークンについては既に 5.2 節で述べた 3,204,251 個のトークンに含まれている．

5.5　個人開発者数のカウントに誤りが生じる場合

5.1 節で述べたようにソースコードの作成者が個人のメールアドレスのドメインを使いながら団体の一員として開発をしている場合や，団体のドメインを使いながら個人的に開発している場合には個人開発者数のカウントにも誤りが生じてしまう．したがって 4.2 節の調査結果にも誤りが生じている可能性がある．

6　まとめと今後の課題

本研究では Linux カーネルにおける著作権表示の有無に関してトークン単位で調査した．まず，著作権表示があるトークンの割合とすべてのトークンに著作権表示があるファイルの割合を求めた．その結果，著作権表示があるトークンの割合は十分とは言えないこと，すべてのトークンに著作権表示があるファイルの割合は非常に低いということがわかった．さらに，同様の調査を過去のバージョンの Linx カーネルに対しても行った．結果を比較したところ，それらの割合は改変を経て低下していることがわかった．また，著作権表示があるトークンと著作権表示がないトークンがどのような場合に発生するのかを調べるために 2 つの視点から調査を行なった．まず，団体での開発と個人での開発に注目し，それぞれの開発で追加されたトークンのうち著作権表示があるトークンの割合に違いがあるのかを調査した．結果として，団体での開発の方が個人での開発によりも著作権表示があるトークンの割合がやや高かったものの，大きな違いは見られなかった．また，団体別の分布と個人開発者別の分布もよく似ており，著作権表示があるトークンの割合が低い団体と個人開発者が最も多かった．さらに，そのような著作権表示があるトークンの割合が低い団体や個人について調べたところ団体での開発か個人での開発かによらず追加したトークン数が少ない団体と個人開発者が最も多かった．次に，Linux カーネルの機能別に著作権表示があるトークンの割合に関して調査した．その結果，各機能ごとに大きな違いが見られた．しかし，その違いが生まれた原因を探るため，いくつかの著作権表示とは直接の関連がない指標との相関を調べたが明確な相関があるものはなかった．

よって今後の課題として，機能別に結果が大きく分かれた原因を調べ著作権表示があるトークンの割合が高くなる要因や低くなる要因を知ること，ファイルやトークンに対して団体での開発か個人での開発かよる分類と機能別の分類以外の分類を行うことで著作権表示があるトークンの発生原因を探ること，また，4.1 節で述べたようなソースコードが変更を加えられる際に，変更による著作権表示がないトークンの新たな発生を検知して警告を行うシステムを作成することがあげられる．

謝辞

大阪大学大学院情報科学研究科コンピュータサイエンス専攻井上克郎教授には，非常に多忙な中，研究において大変貴重な御指導および御助言を賜りました．井上教授から多く賜った適切な御指導により，本論文を完成させることができました．井上教授に心より深く感謝いたします.

大阪大学大学院情報科学研究科コンピュータサイエンス専攻松下誠准教授には，研究の各段階において多くの御助言を賜りました．多くの御指導および御助言を頂いた松下准教授に心より深く感謝いたします.

大阪大学大学院情報科学研究科コンピュータサイエンス専攻春名修介特任教授には研究室での発表の機会において多くの御意見・御助言を賜りました．春名特任教授に心より深く感謝いたします.

大阪大学大学院情報科学研究科コンピュータサイエンス専攻神田哲也助教には，ミーティングの際や発表練習の際に大変貴重な御意見を賜りました．多くの御助言を頂いた神田助教に心より深く感謝いたします.

大阪大学大学院情報科学研究科コンピュータサイエンス専攻仇実氏には，研究方針や内容に関して多くの指導を賜りました．不慣れな私に研究について教えてくださった仇氏に心より深く感謝いたします.

井上研究室の先輩方におきましては，たくさんの研究に関する御助言をいただきました. 特に，大阪大学大学院情報科学研究科コンピュータサイエンス専攻 伊藤薫氏，嶋利一真氏，栗原拓己氏には，研究や発表内容に関する助言，添削など，様々な場面でご協力していただきました. 先輩方の御指導のおかけで本論文を完成させることかできました，心より深く感謝いたします.

最後に，その他様々な御指導，御助言等をいただいた大阪大学大学院情報科学研究科コンピュータサイエンス専攻井上研究室の皆様に感謝いたします.

参考文献

[1] S. Chacon and B. Straub. Pro git. Vol. 515, pp. 13–17, 2020.

[2] Oracle Corporation. The oracle contributor agreement. https://www.oracle.com/technical-resources/oracle-contributor-agreement.html.

[3] M. Di Penta and D. M. German. Who are source code contributors and how do they change? In *2009 16th Working Conference on Reverse Engineering*, pp. 11–20, 2009.

[4] D. M. German, B. Adams, and K. Stewart. Cregit: Token-level blame information in git version control repositories. *Empirical Software Engineering*, Vol. 24, No. 4, 2019.

[5] Inc. GitHub. Where the world builds software. https://github.com/git-guides.

[6] B. Gobeille and H. Packard. The fossology project. https://www.linuxfoundation.org/wp-content/uploads/lfcorp/files/lf_foss_compliance_fossology.pdf.

[7] Open Source Initiative. The open source definition. https://opensource.org/docs/osd.

[8] H. Meeker. Patrick mchardy and copyright profiteering. https://opensource.com/article/17/8/patrick-mchardy-and-copyright-profiteering.

[9] The World Trade Organization. Trips 協定. https://www.jpo.go.jp/system/laws/gaikoku/trips/index.html.

[10] World International Property Organization. Crnr/dc/94 wipo copyright title, 1996. .

[11] S. Qiu, D. M. German, and K. Inoue. An exploratory study of copyright inconsistency in the linux kernel. *IEICE Transactions on Information and Systems*, Vol. E104.D, No. 2, pp. 254–263, 2021.

[12] K. Stewart, S. Khan, and D.M. German. 2020 linux kernel history report. https://www.linuxfoundation.jp/wp-content/uploads/2020/08/2020_kernel_history_report_082720.pdf.