TEX Live ガイド 2024

Karl Berry 編
朝倉卓人（wtsnjp） 訳

https://tug.org/texlive/

2024 年 3 月

目次

1 イントロダクション 5
1.1 TEX Live と TEX コレクション 5
1.2 サポート OS 5
1.3 TEX Live のインストール（基本） 6
1.4 セキュリティについて 6
1.5 サポート情報 7

2 TEX Live の概要 9
2.1 TEX コレクション: TEX Live, MiKTEX, MacTEX 9
2.2 TEX Live のトップレベルディレクトリ 10
2.3 いろいろな TEXMF ツリー（概要） 11
2.4 TEX の拡張エンジン 13
2.5 TEX Live に含まれるその他の著名なソフトウェア 15

3 TEX Live のインストール 16
3.1 インストーラの入手と起動 16
3.1.1 Unix 系 17
3.1.2 macOS 18

3.1.3 Windows 19
3.1.4 Cygwin 19
3.1.5 テキストモード 20
3.1.6 GUI モード 21
3.1.7 その他のレガシーなモード 21
3.2 インストーラの操作方法 22
3.2.1 バイナリ選択メニュー（Unix 系のみ） 22
3.2.2 スキーム・コレクションの選択 23
3.2.3 インストール先ディレクトリ 24
3.2.4 オプション 26
3.3 install-tl のコマンドラインオプション 28
3.3.1 -repository オプション 29
3.4 インストール後のアクション 30
3.4.1 Unix 系における環境変数の設定 30
3.4.2 環境変数をグローバルに設定する 31
3.4.3 DVD インストール後のインターネットを利用したアップデート . . . 31
3.4.4 XeTeX と LuaTeX のためのシステム設定 32
3.4.5 ConTeXt LMTX と MKIV 33
3.4.6 ローカルおよび個人用のマクロを利用する 34
3.4.7 サードパーティフォントを利用する 35
3.5 インストールした TeX Live をテストする 35
3.6 アンインストール 38
3.7 その他のダウンロード可能なソフトウェア 39

4 特殊なインストール 40
4.1 共有インストール 40
4.2 ポータブル（USB）インストール 41

5 tlmgr: TeX Live マネージャ 42
5.1 tlmgr の GUI 43
5.2 コマンド使用例 44

6 Windows 向けの情報 46
6.1 Windows 専用の機能 46

6.2 Windows 向けに追加されているソフトウェア 47
6.3 外部の Perl, Tcl/Tk, Ghostscript を利用する 48
6.4 ユーザプロファイルがホームディレクトリ扱い 48
6.5 レジストリ 49
6.6 パーミッション 49
6.7 Windows と Cygwin でメモリ上限を増やす方法 50

7 Web2C ユーザガイド 50
7.1 Kpathsea パス検索 52
7.1.1 パスの設定元 53
7.1.2 設定ファイル 53
7.1.3 パス展開 55
7.1.4 デフォルト展開 56
7.1.5 ブレース展開 56
7.1.6 サブディレクトリ展開 57
7.1.7 `texmf.cnf` で使用できる特殊文字の概要 57
7.2 ファイル名データベース 58
7.2.1 ファイル名データベース 58
7.2.2 `kpsewhich`: パス検索用コマンドラインツール 59
7.2.3 使用例 60
7.2.4 デバッグアクション 63
7.3 ランタイムオプション 66
7.4 `$TEXMFDOTDIR` 67

8 謝辞 67

9 更新履歴 71
9.1 過去 71
9.1.1 2003 73
9.1.2 2004 73
9.1.3 2005 77
9.1.4 2006–2007 78
9.1.5 2008 79
9.1.6 2009 80

texlive-ja (2024-03-10 11:49)

9.1.7 2010 81
9.1.8 2011 82
9.1.9 2012 83
9.1.10 2013 84
9.1.11 2014 85
9.1.12 2015 86
9.1.13 2016 87
9.1.14 2017 88
9.1.15 2018 90
9.1.16 2019 91
9.1.17 2020 92
9.1.18 2021 94
9.1.19 2022 97
9.1.20 2023 99
9.2 現在：2024 102
9.3 未来 103

訳者まえがき：日本語版『TEX Live ガイド』について

この文書は毎年更新される TEX Live の公式ドキュメント *The TEX Live Guide* を個人が翻訳したものです。内容としては TEX Live についてのごく初歩的な知識、インストール方法、TEX Live マネージャを用いた TEX Live 管理方法が順番に解説され、さらに後半では Kpathsea ライブラリの基礎や今日までの TEX Live における主な変更の履歴など、少し高級な事柄までが扱われています。

TEX Live ガイドは日本語を含め約 10 ヶ国語に翻訳されていることからもわかるように、欧米圏のユーザのみならず全世界の TEX Live ユーザ向けの文書です。しかし、基礎的な解説の中で言及されている TEX 関連プログラムは主として欧文向けのものに限られ、この日本語版 TEX Live ガイドの読者がおそらく必要としているだろう、TEX で和文を処理する際に必要な情報は不足している部分があることを否めません。そのような情報を本翻訳ドキュメントだけで完全に補完することは難しいですが、訳者が可能な範囲で、日本語ユーザにとって有益と思われる補足情報をいくつかの「日本語 TEX ユーザ向けのノート」コラムや訳注の形で追加しています。

本翻訳ドキュメントについて、訳者は慎重な翻訳と技術的に正確な記述を心がけていま

すが、一部に不正確な翻訳や技術的に誤った記述が含まれている可能性があります。本ドキュメントの内容に基づく運用結果については責任を負いかねますので、ご了承ください。また具体的に翻訳が不適切な箇所等を見つけられましたら、ぜひ本翻訳プロジェクトの GitHub リポジトリまでご報告をお願いします：

https://github.com/wtsnjp/texlive-ja

最後に、本翻訳ドキュメントの作成にあたって八登崇之氏から多数のフィードバックをいただきました。この場を借りて深く御礼申し上げます。

1 イントロダクション

1.1 TEX Live と TEX コレクション

このドキュメントは GNU/Linux やその他の Unix 系システム、macOS、Windows 向けに TEX 関連プログラムを集めたディストリビューション TEX Live の概要について解説するものです。

この文書の読者は既にインターネット、TEX Users Group の配布する TEX コレクション DVD またはその他の方法を利用して TEX Live を入手していることと思います。2.1 節は TEX コレクション DVD の収録物について簡単に説明しています。TEX Live と TEX コレクションは TEX Users Group（TUG）の協力によって維持されています。このドキュメントは主として TEX Live それ自体を解説するものです。

TEX Live には TEX, LATEX 2_ε, ConTEXt, METAFONT, METAPOST, BIBTEX 実行プログラムはもちろん、TEX を拡張する多数のマクロパッケージやフォント、そしてそれらの付属ドキュメントが含まれています。また、世界中で利用されるさまざまな言語における組版もサポートされています。

TEX Live のバージョンごとの主な変更点については、このドキュメントの末尾にある 9 節（p. 71）を参照してください。

1.2 サポート OS

TEX Live には GNU/Linux, macOS, Cygwin を含む、多くの Unix 系プラットフォーム向けのバイナリが含まれています。同梱されているソースを用いれば、デフォルトではサポートされていないプラットフォームでコンパイルを行うこともできます。

Windows については、Windows 7 以上のバージョンをサポートしています。いまのと

ころ Windows Vista でも概ね動作する可能性がありますが、Windows XP よりも古いものについては TEX Live のインストールすら成功しません。TEX Live は Windows 向けには 64-bit バイナリを収録しています。

Windows および macOS に対する TEX Live 以外のディストリビューションについては 2.1 節を参照してください。

1.3 TEX Live のインストール（基本）

TEX Live は DVD またはインターネットを利用してインストールすることができます（https://tug.org/texlive/acquire.html）。インターネット・インストーラ自体はとても小さなプログラムで、これを実行すると必要とされるすべてのファイルをインターネットからダウンロードします。

DVD インストーラはあなたのローカルディスクに TEX Live をインストールするものですが、TEX コレクション DVD 上にある TEX Live のデータ（またはその .iso イメージ）を直接実行することはできません。USB ディスクなどにポータブルインストールを行うことも可能です（4.2 節を参照）。インストールについては後ほど詳述しますが（p. 16）、ここではクイックスタートとして簡単に説明します：

- Unix 系ではインストールスクリプトは install-tl です。Windows では代わりに install-tl-windows を実行する必要があります。このプログラムに -gui オプションを与えると「GUI モード」（Windows でデフォルト）、-gui=text を与えると「テキストモード」（その他のプラットフォームでデフォルト）が実行されます。
- TEX Live をインストールすると「TEX Live マネージャ」（tlmgr）も利用可能になります。インストーラと同様、このプログラムにも GUI モードとテキストモードがあります。TEX Live マネージャを用いてパッケージのインストールやアンインストール、その他さまざまな設定タスクを実行することができます。

1.4 セキュリティについて

私たちの知る限り、TEX の主要なプログラムそれ自体はとても堅牢です（今までもずっとそうでした）。しかしながら、TEX Live に含まれる多数のサードパーティ製プログラムは、多くの人の尽力があるとはいえ、それでも同じレベルには達していないかもしれませ

ん。一般論ですが、信頼できない入力についてプログラムを実行する際は十分に注意してください。なるべく安全に実行するためには、新しいサブディレクトリを作って実行したり、chroot を利用したりしてください。

この注意は Windows については特に重要です。なぜなら Windows は、検索パスの設定に関わらず、常にカレントディレクトリにあるプログラムを他の何よりも優先して実行するからです。この挙動はさまざまな攻撃に悪用される恐れがあります。我々は多くのセキュリティホールを塞いできましたが、特にサードパーティ製プログラムについては、間違いなくまだ残っているものが数多くあるでしょう。したがって、カレントディレクトリにある疑わしいファイル（特にバイナリとスクリプトファイル）は事前にチェックしておくことを推奨します。悪意のあるファイルはそもそも存在するべきでありませんが、単にドキュメントを処理するだけで生成されることなどあってはなりません。

最後に、TEX（およびその周辺ツール）はドキュメントの処理中にファイル書き込みを行うことができますが、その機能もまたさまざまな方法で利用され得ます。再度の注意になりますが、ご自身で作成されたわけではないドキュメントは、必ず新しいサブディレクトリで実行するのが無難です。

セキュリティに関する別の観点として、ダウンロードしたファイルが元々作成されていたものから改変されていないかどうかということがあります。tlmgr（5 節）は gpg（GNU Privacy Guard）が利用可能な場合、ダウンロードしたファイルについて自動的に暗号理論に基づく検証を行います。gpg は TEX Live に含まれていませんが、このプログラムについての情報は https://texlive.info/tlgpg/ に記載してありますので必要に応じて参照してください。

1.5 サポート情報

TEX コミュニティはアクティブかつ友好的で、ほとんどの真剣な質問には回答を得ることができます。しかしながら、こうしたサポートは非公式で、ボランティアや一般の TEX ユーザ有志によって行われているものですから、質問をする前にドキュメントや過去の Q&A をよく読んで、まずはご自分で解決できるよう最善を尽くしましょう（もし保証付きの有償サポートをご希望の場合は、TEX Live の採用は諦めて商用のシステムをご利用ください。https://tug.org/interest.html#vendors には TEX に関連する商用サービスを提供する団体の一覧があります）。

以下に TEX 関連の情報源をリストアップしておきます。順番は概ねおすすめ順です：

はじめての方へ これからはじめて TEX をお使いになる場合は https://tug.org/begin.html にアクセスするとよいでしょう。TEX システムについて簡潔なイントロダクションを読むことができます。

CTAN 特定の TEX パッケージやフォント、プログラム等を見つけたい場合はまず CTAN で探すとよいでしょう。これはあらゆる TEX 関連の成果物を集めた巨大なコレクションです。CTAN カタログの個別ページでは、見つかったパッケージが TEX Live または MiKTEX に含まれるかどうかの情報も提供されています。https://ctan.org/pkg/catalogue/ にアクセスしてみてください。

TEX FAQ TEX に関するよくある FAQ を集めた巨大なデータベースで、初歩的な質問からとても難解な質問まで含まれています。TEX FAQ はインターネット経由で利用することができます（https://texfaq.org/）。

ウェブ上の情報源 ウェブサイト https://tug.org/interest.html には多数の TEX 関連リンクが集められています。特に TEX システムに関わるさまざまな話題についての多くの書籍、マニュアル、投稿記事を多数紹介しています。

サポートサイト まずは通常のウェブ検索を試すことをおすすめします。TEX に関する主要なサポートサイトとしては LATEX コミュニティ https://latex.org や Q&A サイト https://tex.stackexchange.com、Usenet ニュースグループ news:comp.text.tex、メーリングリスト texhax@tug.org などが挙げられます。これらのアーカイブには何年分もの質問と回答が蓄積されていますので、探してみるとよいでしょう。

質問する もし答えの見つからない質問にぶつかったときは、https://latex.org や https://tex.stackexchange.com/ のウェブインターフェースや、メーリングリスト texhax@tug.org（購読せずに投稿して構いません）を通じて質問することもできます。ただし、どこで質問を投稿するにしても、必ず事前に「質問のしかた」（https://texfaq.org/FAQ-askquestion）を読むようにしましょう。この内容に従うことで、有効な回答を得られる可能性を飛躍的に高めることができます。

TEX Live サポート もし TEX Live ディストリビューションに関してバグ報告や提案、コメントがある場合は TEX Live メーリングリスト tex-live@tug.org にメールを投稿してください。ただし TEX Live に含まれる個別のプログラムの使い方についての質問等は、TEX Live メーリングリストではなく、各プログラムのメンテナや専用のメーリングリストに連絡するようにしてください。多くの場合、プログラムを `--help` オプション付きで実行するとバグ報告用のメールアドレスを確認することができます。

日本語 TEX ユーザ向けノート

上記リストは主に英語での情報提供・コミュニケーションを行う場所ですが、日本語でのサポートが必要な場合は、次のようなウェブサイトが利用できます：

TEX Wiki https://texwiki.texjp.org/ は日本語で読めるものとしては TEX 関連情報を集めた最大の情報源です。現在は日本語 TEX 開発コミュニティ（texjporg）がサーバ管理を行っていますが、あくまで Wiki なので編集は誰でも行うことができ、掲載情報は必ずしも公式のものではありません。

TEX Forum https://okumuralab.org/tex/ は歴史の長い Q&A サイトで、現在も日本語で TEX に関連する質問ができる場所としては最もアクティブユーザが多く、熟練の TEX ユーザが多数参加しています。上記 TEX Wiki に日本語版の「質問のしかた」（https://texwiki.texjp.org/?質問のしかた）ページがありますので、初めて質問を投稿する際は必ず目を通すようにしましょう。

スタック・オーバーフロー https://ja.stackoverflow.com/ は TEX に限らずあらゆる技術・プログラミング関連の質問を投稿することができる Q&A サイト Stack Overflow の日本語版です。最近はここの LATEX タグ（https://ja.stackoverflow.com/questions/tagged/latex）にも少しずつ TEX/LATEX 関連の質問が投稿されるようになってきています。

TEX ユーザは上記のような場所でサポートを受けることができる一方で、あなた自身が他の質問者の手助けをすることも可能です。いずれのコミュニティも全世界に開かれたものですから、ぜひお気軽に参加・購読し、ヘルプが必要なユーザを助けてあげましょう。

2 TEX Live の概要

このセクションでは TEX Live と TEX コレクションの収録物について紹介します。

2.1 TEX コレクション: TEX Live, MiKTEX, MacTEX

TEX コレクション DVD には以下のものが含まれています：

TEX Live ディスクにインストールするための包括的でクロスプラットフォームな TEX ディストリビューションです。ウェブサイト：https://tug.org/texlive/

MacTEX macOS 向けの TEX システムで、TEX Live に専用のインストーラといくつかの Mac 用アプリケーションを追加しています。ウェブサイト：https://tug.org/mactex/

MiKTEX これも包括的でクロスプラットフォームな TEX ディストリビューションです。Windows, GNU/Linux, macOS をサポートしていますが、DVD に含まれているのは Windows 向けのバイナリのみです。MiKTEX は専用のパッケージマネージャを備えており、必要なコンポーネントを必要に応じてインターネットからインストールするようになっています。ウェブサイト：https://miktex.org/

CTAN CTAN リポジトリ（https://ctan.org/）のスナップショットです。CTAN は TEX Live と同じライセンスで配布されているわけではないので、再配布や変更を行う場合は十分注意してください。

2.2 TEX Live のトップレベルディレクトリ

ここに TEX Live ディストリビューションのトップレベル（最上位階層）にあるディレクトリの一覧を示し、それぞれについて簡単に説明しておきます。

`bin` TEX 関連プログラムを格納しています。実行バイナリはプラットフォームによって異なります。

`readme-*.dir` TEX Live の概要と有用なリンクを集めた多言語の README を含んでいます。HTML バージョンとテキストバージョンがあります。

`source` TEX Live に含まれるすべてのプログラムのソースコードです。Web2C で記述された、TEX ディストリビューションのコアプログラムのソースコードも含まれています。

`texmf-dist` 最も重要な TEXMF ツリーです。詳しくは下にある `TEXMFDIST` の項目を参照してください。

`tlpkg` インストールに必要なスクリプトや Windows 向けの補助プログラムを含んでいます。

ドキュメントを探す際には、同じくトップレベルにある `doc.html` が役立ちます。このファイルには `texmf-dist/doc` 以下に含まれるほとんどすべてのドキュメント（パッケージ、フォーマット、フォント、プログラムのマニュアルや man ページなど）が掲載されています。また TEX Live に含まれるドキュメントを検索するには `texdoc` コマンド

を利用することもできます。

この TEX Live ガイド自体は texmf-dist/doc/texlive 以下にあり、多数の言語に翻訳されています：

- チェコ・スロバキア語：texmf-dist/doc/texlive/texlive-cz
- ドイツ語 texmf-dist/doc/texlive/texlive-de
- 英語：texmf-dist/doc/texlive/texlive-en
- フランス語：texmf-dist/doc/texlive/texlive-fr
- イタリア語：texmf-dist/doc/texlive/texlive-it
- 日本語：texmf-dist/doc/texlive/texlive-ja
- ポーランド語：texmf-dist/doc/texlive/texlive-pl
- ロシア語：texmf-dist/doc/texlive/texlive-ru
- セルビア語：texmf-dist/doc/texlive/texlive-sr
- スペイン語：texmf-dist/doc/texlive/texlive-es
- 簡体字中国語：texmf-dist/doc/texlive/texlive-zh-cn

2.3 いろいろな TEXMF ツリー（概要）

この節では、システムに利用される種々の TEXMF ツリーを指定するために予め定義されている変数を一覧にし、それらの所期の目的および TEX Live でのデフォルト構成を説明します。コマンド tlmgr conf を実行するとこれらの変数の値が表示されるので、各 TEXMF ツリーがあなたのコンピュータで実際にどのディレクトリを指し示すのか容易に確認することができます。

個人用のものも含め、すべての TEXMF ツリーはその大量のサブディレクトリとともに TEX ディレクトリ構成（TDS, https://tug.org/tds）にしたがうべきです。TDS にしたがわない構成の場合、ファイルが見つからない可能性があります。詳細については 3.4.6 節（p. 34）を参照してください。ここで列挙するのはツリーが検索される順序の逆順です。つまり、より後に登場するツリーによって、それより前のツリーの内容は上書きされます。

TEXMFDIST このツリーには TEX Live ディストリビューション自体に含まれるほとんどのファイル（設定ファイル、スクリプト、パッケージ、フォント、etc.）が配置されています。（主な例外はプラットフォーム毎の実行バイナリで、それらはトップ

レベルの bin ディレクトリに配置されています。)

TEXMFSYSVAR この（システム用の）ツリーは texconfig-sys, updmap-sys, fmtutil-sys, tlmgr などが、フォーマットや生成された map ファイルなどの実行時（キャッシュ）データを保存するためのものです。

TEXMFSYSCONFIG この（システム用の）ツリーは texconfig-sys, updmap-sys, fmtutil-sys などが、変更された設定データを保存するために利用するものです。

TEXMFLOCAL このツリーは、システム管理者が全ユーザに適用するためにマクロやフォントなどを追加または更新して配置するためのものです。

TEXMFHOME このツリーは、一般のユーザが独自に追加または更新したいマクロやフォントを配置するためのものです。この変数の参照先は、使用するユーザごとに固有のディレクトリです。

TEXMFVAR この（ユーザ向けの）ツリーは texconfig, updmap-user, fmtutil-user などがフォーマットや生成された map ファイルなどの実行時（キャッシュ）データを保存するためのものです。

TEXMFCONFIG この（ユーザ向けの）ツリーは texconfig, updmap-user, fmtutil-user などが変更された設定データを保存するために利用するものです。

TEXMFCACHE このツリーは ConTEXt MkIV および LuaLATEX が実行時（キャッシュ）データを保存するためのものです。デフォルトでは TEXMFSYSVAR（またはこれが書き込み不能な場合は TEXMFVAR）に設定されています。

デフォルトの TEX Live 構成：

システム用ルート 複数の TEX Live を配置できます（デフォルト値は Unix 系では /usr/local/texlive）

2023 前年のリリース

2024 最新のリリース

bin

i386-linux GNU/Linux 向けバイナリ (32-bit)

⋮

universal-darwin macOS 向けバイナリ

x86_64-linux GNU/Linux 向けバイナリ (64-bit)

windows Windows 向けバイナリ (64-bit)

texmf-dist TEXMFDIST, TEXMFMAIN

texmf-var TEXMFSYSVAR, TEXMFCACHE

texmf-config TEXMFSYSCONFIG

texmf-local TEXMFLOCAL（リリースに依存しません）

ユーザのホームディレクトリ（$HOME または %USERPROFILE%）

.texlive2023 前年リリースに適用するユーザごとのデータ（設定や生成物）

.texlive2024 最新リリースに適用するユーザごとのデータ（設定や生成物）

texmf-var TEXMFVAR, TEXMFCACHE

texmf-config TEXMFCONFIG

texmf TEXMFHOME（ユーザごとのマクロなどを配置）

2.4 TEX の拡張エンジン

Knuth によるオリジナルの TEX は既に開発が終了しており、ごく稀にバグ修正が入る程度です。TEX Live には現在もこのオリジナル処理系が tex として含まれており、この状況は当面の間は変わることはないでしょう。一方で、TEX Live にはいくつかの TEX を拡張したエンジン（それらも TEX 処理系と呼ばれています）も収録されています：

ε-TEX この処理系ではいくつかの追加プリミティブ（主としてマクロ展開、文字列スキャン、\marks クラス、デバッグに関わるもの）と双方向組版のための TEX--XETT 拡張が利用可能です。デフォルトモードでは、ε-TEX はオリジナルの TEX と完全な互換性があります。詳細については texmf-dist/doc/etex/base/etex_man.pdf を参照してください。

pdfTEX この処理系には、ε-TEX の機能に加えて DVI 同様に PDF を出力する機能と、その他出力とは関係のない多数の追加機能が搭載されています。この処理系は、etex, latex, pdflatex など多くの一般的なフォーマット（コマンド）から呼び出されています。pdfTEX のウェブサイトは https://tug.org/applications/pdftex/ です。TEX Live にはマニュアル（texmf-dist/doc/pdftex/manual/pdftex-a.pdf）と pdfTEX の拡張機能の利用例を示した文書（texmf-dist/doc/pdftex/samplepdftex/samplepdf.tex）も収録されています。

LuaTEX Unicode による入力と OpenType/TrueType フォントおよびシステムフォントのサポートを追加した処理系です。この TEX エンジンには軽量スクリプト言語 Lua（https://www.lua.org/）のインタプリタが組み込まれており、TEX における厄介な問題に対処するのにとても役立ちます。texlua として呼び出した場合は

独立の Lua 処理系のように振る舞います。ウェブサイト：https://luatex.org/、マニュアル：texmf-dist/doc/luatex/base/luatex.pdf。

(ε-)(u)pTEX 日本語組版をネイティブにサポートする処理系です。pTEX がベースですが、ε-TEX 拡張が利用可能な ε-pTEX と Unicode サポートを含む upTEX もあります（詳細はこのリストの下のコラムを参照）。

XƎTEX Unicode による入力と OpenType/TrueType フォントおよびシステムフォントの利用をサードパーティ製のライブラリを利用してサポートする TEX 処理系です。ウェブサイト：https://tug.org/xetex。

Ω (Omega) Unicode（16-bit 文字）ベースの処理系で、すなわち世界中のほとんどあらゆる言語で用いられる文字に対応しています。この処理系はいわゆる Omega Translation Processes（OTPs）もサポートしており、任意の入力について複雑な組版処理を施すことも可能です。

ℵ (Aleph) Omega と ε-TEX を組み合わせた処理系です。詳細については texmf-dist/doc/aleph/base を参照してください。

日本語 TEX ユーザ向けノート

日本語組版向けに開発された pTEX 系の処理系を除いては、欧文組版を行うために開発されたもので、多くがそのままでは日本語組版には適しません。TEX Live に含まれる処理系の中で、日本語組版に適したものを以下に列挙します。

pTEX Knuth によるオリジナルの TEX 処理系を日本語組版用に拡張したものです。和文フォントを扱うことができるように拡張されているほか、禁則処理や縦組みにも対応しています。詳細については texmf-dist/doc/ptex/ptex-manual/ptex-manual.pdf を参照してください。

upTEX pTEX の派生エンジンで、内部コードが Unicode に変更されています。基本的には upTEX は pTEX の上位互換となっており、UTF-8 入力が活用できるのみならず、他にもいくつかの機能が追加されています。

ε-pTEX, ε-upTEX pTEX と upTEX に ε-TEX 相当の拡張機能を追加したエンジンです。現在 TEX Live に含まれる platex, uplatex コマンドではデフォルトでこれらの処理系が呼び出されます。最近は pdfTEX 由来の機能も一部利用できます。詳細は texmf-dist/doc/ptex/ptex-base/eptexdoc.pdf を参照してください。

また LuaTEX-ja というマクロパッケージを用いると LuaTEX でも日本語組版を実現できます。詳しくはウェブサイト https://ja.osdn.net/projects/luatex-ja/ またはドキュメント texmf-dist/doc/luatex/luatexja/luatexja-ja.pdf を参照してください。

2.5 TEX Live に含まれるその他の著名なソフトウェア

ここに少しですが、TEX Live に含まれていて、よく利用されるプログラムを列挙しておきます：

bibtex, biber　参考文献リストの作成を補助するプログラム。

makeindex, xindex, xindy　索引の作成を補助するプログラム。

dvips　DVI を PostScript に変換するプログラム。

dvipdfmx　DVI を PDF に変換するプログラム。先述の pdfTEX とは別のアプローチを提供するもの。

xdvi　X Windows システム向けの DVI ビューア。

dviconcat, dviselect　DVI のページを切り貼りするためのプログラム。

psselect, psnup, ...　PostScript ユーティリティ。

pdfjam, pdfjoin, ...　PDF ユーティリティ。

context, mtxrun　ConTEXt と PDF プロセッサ。

htlatex, ...　tex4ht は (LA)TEX を HTML, XML, DocX などに変換するプログラム。

日本語 TEX ユーザ向けノート

日本では、上記に加えて次のプログラムもよく用いられます：

pbibtex, upbibtex　bibtex を日本語用に拡張したもの。

mendex, upmendex　makeindex を日本語用に拡張したもの。

3 TEX Live のインストール

3.1 インストーラの入手と起動

TEX Live をインストールするためには、まず TEX コレクション DVD を入手するか TEX Live のインターネット・インストーラをダウンロードします。より詳細な情報や TEX Live を入手するその他の方法については https://tug.org/texlive/acquire.html を参照してください。

インターネット・インストーラ（.zip または .tar.gz） CTAN の systems/texlive/tlnet 配下からダウンロードできます。http://mirror.ctan.org/systems/texlive/tlnet にアクセスすると、最寄りの最新ミラーにリダイレクトされるはずです。install-tl.zip（Unix 系および Windows 向け）、またはそれよりもかなりファイルサイズの小さい install-unx.tar.gz（Unix 系専用）のお好きな方をダウンロードしてください。ダウンロードしたアーカイブを展開すると install-tl サブディレクトリ以下に install-tl および install-tl-windows.bat が見つかるはずです。

インターネット・インストーラ（Windows 用 .exe） 上と同様に CTAN からダウンロードし、ダブルクリックしてください。すると、インストーラの最初の画面（図 1）が表示されるはずです。この画面では「インストール」または「展開のみ」のいずれかを選ぶことができます。

TEX コレクション DVD DVD 内の texlive サブディレクトリを開いてください。Windows では、通常 DVD を挿入するとインストーラが自動的に起動するはずです。TEX Users Group（TUG）の会員になるか（https://tug.org/usergroups.html）、TUG のオンラインストア（https://tug.org/store）で個別に購入すると DVD を手に入れることができます。また、CTAN で配布されている ISO イメージからご自身で DVD に焼くことも可能です。DVD または ISO から TEX Live をインストールした後、インターネットを利用して継続的にアップデートを行いたい場合は 3.4.3 節を参照してください。

いずれの手段でインストーラを入手しても、まったく同じプログラムが起動します。入手方法の違いによって生じる差異のうち、ユーザにとって最も重要なことは、インターネット・インストーラを利用した場合はすべてのパッケージについて入手可能な最新バー

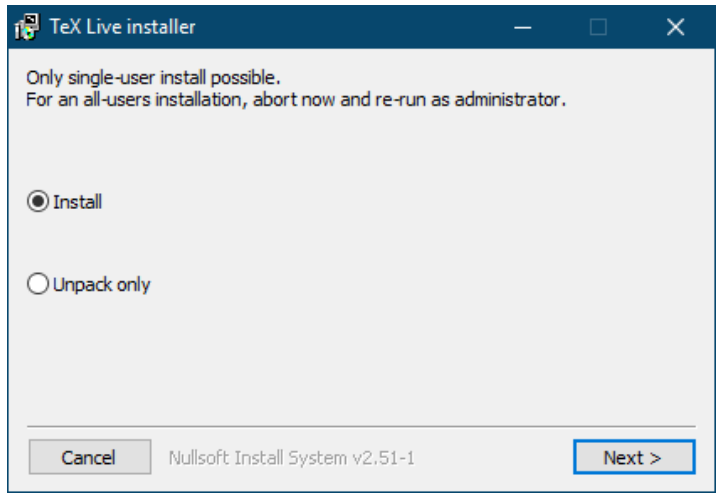


図 1　Windows 用インストーラ（.exe）の初期画面。

ジョンがインストールされますが、DVD や ISO イメージを用いた場合には年に一度のパブリックリリース時点での最新版がインストールされるということです。

もしプロキシを利用してダウンロードする必要がある場合は、~/.wgetrc ファイルまたは環境変数を利用して Wget に対して適切なプロキシ設定を行うか（https://www.gnu.org/software/wget/manual/html_node/Proxies.html）、もしくはお好みで他のダウンロード用ツールを利用してください。DVD や ISO イメージからインストールを行う場合は、プロキシ設定について気にする必要はありません。

以降のセクションでは、このインストーラの使い方について詳しく説明します。

3.1.1　Unix 系

以降では > はシェル・プロンプトを表し、ユーザの入力はタイプライタ体で command のように表現することにします。TEX Live のインストーラ install-tl は Perl スクリプトです。Unix 系システムでこのスクリプトを起動する最も簡単な方法は以下を実行することです[*1]：

```
> perl /path/to/installer/install-tl
```

ところで、インストーラのメッセージを見やすく表示するためには、ターミナルのウィン

[*1] もちろん、実行権限を与えていれば単に /path/to/installer/install-tl として起動することもできますし、事前に cd で install-tl のあるディレクトリに移動しても構いません。以降では、そのようなバリエーションについて逐一言及はしません。

ドウサイズは十分に大きくしておいてください（図 2）。

インストーラを GUI モードで起動するためには、予め Tcl/Tk をインストールしておく必要があります。その上で、次のようにすると GUI モードで起動できます：

```
> perl install-tl -gui
```

古い -wizard, --perltk, --expert オプションは現在では -gui と等価になっています。利用可能なすべてのオプションを確認するには次を実行してください：

```
> perl install-tl -help
```

■Unix のパーミッションについて TEX Live インストーラは、実行時の umask 設定を反映します。したがって、もし利用中のユーザだけでなく他のユーザも利用可能な形で TEX Live をインストールしたい場合は、umask 002 など適切なパーミッション設定になっているかよく確認してください。umask コマンドの詳細については、システムのマニュアルを参照してください。

■Cygwin への注意 他の Unix 系システムと異なり、Cygwin にはデフォルトでは TEX Live インストーラの実行に必要なプログラムの一部が含まれていません。詳細は 3.1.4 節を参照してください。

3.1.2 macOS

2.1 節でも言及したように、macOS には MacTEX という専用のディストリビューションがあります（https://tug.org/mactex）。MacTEX のインストーラは macOS 向けにいくつかの最適化をしている（たとえば、いわゆる「TEXDist データ構造」を利用して Mac 上の複数の TEX Live リリースを簡単に切り替える機能があります）ので、TEX Live インストーラを利用するよりも、MacTEX を利用したインストールをおすすめします[*2]。

MacTEX は完全な TEX Live ベースのディストリビューションで、主要な TEX ツリーと含まれる実行バイナリはまったく同一です。通常の TEX Live に、macOS 専用のドキュメントとアプリケーションがいくつか追加されています。

[*2] 訳注：ただし敢えて TEX Live インストーラを利用することも可能です。

3.1.3 Windows

もし ZIP ファイルをご自身でダウンロードして展開した場合や DVD を挿入しても自動的にインストーラが起動しなかった場合は、`install-tl-windows.bat` をダブルクリックしてください。

もしくは、コマンドプロンプトを利用してインストーラを起動することも可能です。以下では > はプロンプトを表し、ユーザの入力はタイプライタ体で `command` のように表現することにします。もし、既にインストーラのあるディレクトリにいる場合は単に

```
> install-tl-windows
```

を実行してください。もちろん絶対パスを指定して起動することも可能です。例えば、TEX コレクション DVD を利用していて、その光学ドライブが `D:` であれば：

```
> D:\texlive\install-tl-windows
```

図 3 は GUI インストーラ（Windows ではデフォルト）の初期画面を示しています。

テキストモードでのインストールを行いたい場合は次のようにしてください：

```
> install-tl-windows -no-gui
```

すべてのオプションを表示するには以下を実行してください：

```
> install-tl-windows -help
```

ノート：同じディレクトリに `install-tl-windows.exe` が存在している場合は、起動時に拡張子 `.bat` を付けてください。もっとも、基本的にはそのような状態にはならないはずです例外はローカルに `tlnet` ディレクトリをミラーしている場合です。

TEX Live のインストールパスに非 ASCII 文字を含めることは、必ず避ける必要があります。これは特に、Microsoft ネットワークアカウントの使用を強要される Windows 11 で問題となり得ます。回避策については https://tug.org/texlive/windows.html#nonascii を参照してください。

3.1.4 Cygwin

Cygwin では、TEX Live のインストールを始める前に `setup.exe` プログラムを用いて（もし未導入の場合は）perl と wget をインストールするようにしてください。また以下のパッケージも予めインストールしておくことをおすすめします：

```
Installing TeX Live 2024 from: ...
Platform: x86_64-linux => 'GNU/Linux on x86_64'
Distribution: inst (compressed)
Directory for temporary files: /tmp
...
 Detected platform: GNU/Linux on Intel x86_64

 <B> binary platforms: 1 out of 16

 <S> set installation scheme: scheme-full

 <C> customizing installation collections
     40 collections out of 41, disk space required: 8296 MB (free: 138718 MB)

 <D> directories:
   TEXDIR (the main TeX directory):
     /usr/local/texlive/2024
   ...

 <O> options:
   [ ] use letter size instead of A4 by default
   ...

 <V> set up for portable installation

Actions:
 <I> start installation to hard disk
 <P> save installation profile to 'texlive.profile' and exit
 <H> help
 <Q> quit
```

図 2　TEX Live インストーラのテキストモードでのメイン画面（GNU/Linux）

- fontconfig（XETEX と LuaTEX のため）
- ghostscript（多くの TEX エンジンのため）
- libXaw7（xdvi のため）
- ncurses（インストーラの `clear` コマンドの実行に必要）

3.1.5　テキストモード

図 2 は Unix 系におけるテキストモードのメイン画面を示しています。テキストモードは Unix 系ではデフォルトです。

図 3　TEX Live インストーラの基本画面（macOS）。「高度な設定」ボタンを押すと図 4 のような画面に切り替わります。

このモードは、完全にコマンドラインで完結するもので、カーソルによる操作のサポートはまったくありません。例えば、Tab キーによってチェックボックスや入力フォーム間を移動することはできません。インストール操作はすべて、プロンプトに文字をタイプ（大文字と小文字は区別されます）して Enter キーを押すことによって行います。ターミナルの画面は、入力にしたがって適切に遷移していきます。

テキストモードのインターフェースは、最小限の Perl しかない環境を含めなるべく多くのプラットフォームで動作するようとても原始的に作られています[*3]。

3.1.6　GUI モード

GUI モードはデフォルトではわずかなオプションのみを提供するシンプルな画面でスタートします（図 3）。GUI モードを起動するには

```
> install-tl -gui
```

を実行します。「高度な設定」ボタンを押すとテキストモードとほぼ同程度のオプションを提供する画面に切り替わります（図 4）。

3.1.7　その他のレガシーなモード

古い `perltk` モード、`expert` モード、`wizard` モードは現在も Perl/Tk がインストールされていれば利用可能です。これらはそれぞれインストーラに `-gui=perltk` や

[*3] 訳注：GUI モードは日本語を含む多言語に対応していますが、テキストモードは英語のみです。

図 4 TEX Live インストーラ GUI モードの「高度な設定」画面（macOS）

`-gui=wizard` オプションを与えると起動できます。

3.2 インストーラの操作方法

インストーラは直感的に操作できるように設計されていますが、このセクションでは多様なオプションやサブメニューについて簡単に説明します。

3.2.1 バイナリ選択メニュー（Unix 系のみ）

図 5 はテキストモードのバイナリ選択メニューを示しています。デフォルトではインストーラ実行時に使用中のプラットフォーム向けのバイナリだけがインストールされます。このメニューを利用すると、その他のプラットフォーム向けのバイナリも同様にインストールすることを選択することができます。このメニューは TEX ツリーをさまざまな環境に共通のネットワークで共有する場合やデュアルブートシステムでは有用です。

```
Available platforms:
==============================================================================
   a [ ] Cygwin on x86_64 (x86_64-cygwin)
   b [ ] MacOSX current (10.14-) on ARM/x86_64 (universal-darwin)
   c [ ] MacOSX legacy (10.6-) on x86_64 (x86_64-darwinlegacy)
   d [ ] FreeBSD on x86_64 (amd64-freebsd)
   e [ ] FreeBSD on Intel x86 (i386-freebsd)
   f [ ] GNU/Linux on ARM64 (aarch64-linux)
   g [ ] GNU/Linux on RPi(32-bit) and ARMv7 (armhf-linux)
   h [ ] GNU/Linux on Intel x86 (i386-linux)
   i [X] GNU/Linux on x86_64 (x86_64-linux)
   j [ ] GNU/Linux on x86_64 with musl (x86_64-linuxmusl)
   k [ ] NetBSD on x86_64 (amd64-netbsd)
   l [ ] NetBSD on Intel x86 (i386-netbsd)
   m [ ] Solaris on Intel x86 (i386-solaris)
   o [ ] Solaris on x86_64 (x86_64-solaris)
   p [ ] Windows (64-bit) (windows)
```

図 5　バイナリ選択メニュー

3.2.2　スキーム・コレクションの選択

```
Select scheme:
==============================================================================
 a [X] full scheme (everything)
 b [ ] medium scheme (small + more packages and languages)
 c [ ] small scheme (basic + xetex, metapost, a few languages)
 d [ ] basic scheme (plain and latex)
 e [ ] minimal scheme (plain only)
 f [ ] infrastructure-only scheme (no TeX at all)
 g [ ] book publishing scheme (core LaTeX and add-ons)
 h [ ] ConTeXt scheme
 i [ ] GUST TeX Live scheme
 j [ ] teTeX scheme (more than medium, but nowhere near full)
 k [ ] custom selection of collections
```

図 6　スキーム選択メニュー

図 6 は TEX Live のスキーム選択メニューです。これによって、お好きなスキーム（パッケージコレクションのセット）を選んでインストールを行うことができます。デフォルトでは TEX Live で利用可能なすべてのパッケージを含む full スキームがインストールされます。おすすめはこの full スキームですが、純粋な plain TEX と LATEX だけを使用する

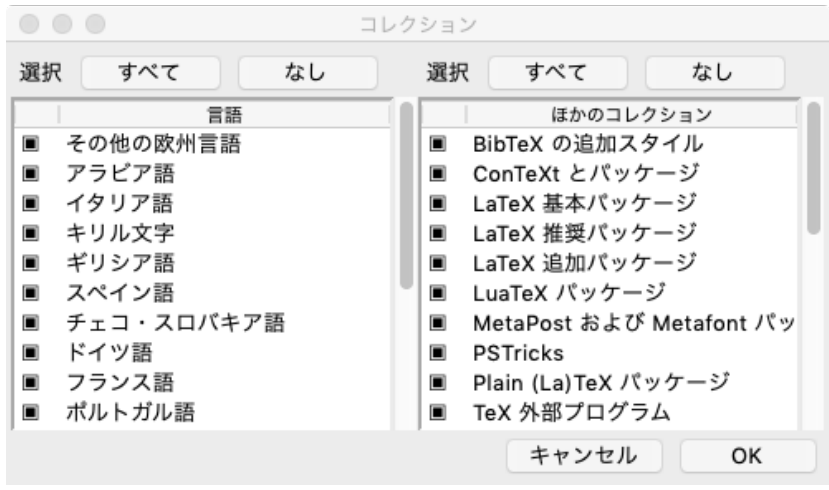


図 7 コレクション選択メニュー

なら basic、それに少しプログラムを追加したものである small（これは MacTEX における BasicTEX と同等です）、単なるテスト目的なら minimal、あるいはそれらの中間にあたるものが欲しければ medium や teTeX スキームを選択することもできます。その他にも、多数の目的別もしくは言語別のスキームが用意されています。

コレクション選択メニュー（図 7）を活用すると選択したスキームをさらに利用目的に応じて最適化することができます[*4]。

コレクションはスキームよりも細かなパッケージのセットです。基本的には、スキームというのはいくつかのコレクションを集めたもので、コレクションは 1 つまたは複数のパッケージをまとめたものです。パッケージは TEX Live におけるプロダクトの最小単位で、具体的には実際の TEX マクロファイルやフォントファイルから成るものです。

もしコレクション選択メニューが提供するよりもさらに細かくインストールするものを制御したい場合は、インストール後に TEX Live マネージャ（`tlmgr`）を使用してください。TEX Live マネージャを利用すればパッケージ単位でインストールを行うことが可能です。

3.2.3 インストール先ディレクトリ

デフォルトの TEX Live 構成については 2.3 節（p. 11）で説明しました。デフォルトのインストール先は Unix 系では `/usr/local/texlive/2024`、Windows では

[*4] 訳注：full スキーム以外のスキームには日本語組版に必要な pTEX 系エンジンや LuaTEX-ja などは含まれていません。したがって full スキーム以外のスキームを選択した場合、日本語組版を行うためには原則として日本語コレクション（collection-langjapanese）を追加する必要があります。

%SystemDrive%\texlive\2024 です。この配置は複数の TEX Live を同時にインストールすることを可能にします。例えば複数のリリース（典型的には、年度ごと）の TEX Live をインストールしておき、単純に検索パスを変更することによってどのリリースを使用するか切り替えるようなことができます。

インストール先のディレクトリはインストーラにおける TEXDIR を設定することによって変更することができます。GUI モードで TEXDIR やその他のオプションを設定するための画面は図 4 に示されています。インストール先を変更する必要があるのは、デフォルトのインストール先にあたるパーティションに十分な空き容量がない場合[*5]や書き込み権限がない場合です[*6]。

Windows では、通常 C:\texlive\2024（より一般的に言えば %SystemDrive%\texlive\2024）を作成するのに管理者権限は必要ありません。

インストール先のディレクトリは、インストーラを実行する前にいくつかの環境変数（よく用いられるのは TEXLIVE_INSTALL_PREFIX と TEXLIVE_INSTALL_TEXDIR）を設定することによっても変更できます。詳細については install-tl --help により表示できるドキュメント（https://tug.org/texlive/doc/install-tl.html）を参照してください。

合理的なインストール場所の変更先は（特にあなたが TEX Live を利用する単独のユーザである場合には）ホームディレクトリ以下でしょう。メタ文字 ~（チルダ）を用いるとこれを簡単に指定することができます（例：~/texlive/2024）。

複数の TEX Live リリースを同時にインストールできるようにするためにも、インストール先のディレクトリ名には「年」を含めることをおすすめします（もし /usr/local/texlive-cur などバージョンに依存しない名前で管理したいとお考えであれば、シンボリックリンクを利用して新しいリリースをテストしてからそのリンク先を変更するようにするのも良いでしょう）。

インストーラにおける TEXDIR を変更すると TEXMFLOCAL, TEXMFSYSVAR, TEXMFSYSCONFIG にあたるディレクトリも合わせて変更されることになります。

TEXMFHOME はユーザごとのマクロファイルやパッケージを配置するのに適した場所で、デフォルトでは ~/texmf です（macOS では ~/Library/texmf）。TEXDIR の場合とは異なり、TEX を実行しているユーザのホームディレクトリを簡単に参照できるように、このデフォルト値では ~ が各々のインストールのために生成された設定ファイルに書き込ま

[*5] TEX Live をフルインストールするには数 GB の容量が必要となります。

[*6] TEX Live をインストールするのに管理者（root）権限は必要ありませんが、インストール先ディレクトリへの書き込み権限は必要です。

れています。~ は Unix 系では $HOME に、Windows では %USERPROFILE% にそれぞれ展開されます。繰り返しの注意になりますが、TEXMFHOME も他の TEXMF ツリーと同様に TDS にしたがった構成になっているべきです（必要のないファイルやディレクトリは配置されていなくも構いません）。

TEXMFVAR はほとんどの（ユーザごとの）実行時キャッシュが保存される場所です。TEXMFCACHE は LuaLATEX と ConTEXt MkIV が使用するための変数で、デフォルトでは TEXMFSYSVAR または（もし TEXMFSYSVAR が書き込み可能でない場合は）TEXMFVAR に設定されています。詳細は 3.4.5 節（p. 33）を参照してください。

3.2.4 オプション

```
Options customization:
==============================================================================
 <P> use letter size instead of A4 by default: [ ]
 <E> execution of restricted list of programs: [X]
 <F> create all format files:                  [X]
 <D> install font/macro doc tree:              [X]
 <S> install font/macro source tree:           [X]
 <L> create symlinks in standard directories:  [ ]
            binaries to:
            manpages to:
                info to:
<Y> after install, set CTAN as source for package updates: [X]
```

図 8 Options menu (Unix)

図 8 はテキストモードにおけるオプションメニューです。以下に各オプションの詳細を列挙します：

use letter size instead of A4 by default（A4 サイズの代わりにレターサイズをデフォルトで使用） デフォルトの用紙サイズを選択します。当然ながら、各ドキュメントのサイズはその都度ドキュメントに合わせて指定することもできます。

execution of restricted list of programs（制限リストにあるプログラムの実行を許可） TEX Live 2010 以降、いくつかの外部プログラムの実行がデフォルトで許可されるようになりました。実行が許可されている（数少ない）プログラムのリストは texmf.cnf に記載されています。詳細については TEX Live 2010 のリリースノート（9.1.7 節）を参照してください。

create all format files（すべてのフォーマットファイルを作成） このオプションは有効に

しておくことを推奨します。このオプションを無効にした場合、各ユーザがフォーマットを使用したいときにフォーマットが動的に生成され、無用な問題を生じる可能性があります。詳細は fmtutil のドキュメントを参照してください。

install font/macro ... tree（フォント・マクロの ～ ツリーをインストール） 多くのパッケージに付属するソースファイルとドキュメントをダウンロード・インストールするか否かを選択するオプションです。このオプションを無効化することは非推奨です。

create symlinks in standard directories（標準ディレクトリにシンボリックリンクを作成） このオプションは Unix 系のみで利用可能で、環境変数の変更を回避する手段です。このオプションを有効にしない場合、TEX Live のディレクトリは通常手動で PATH, MANPATH, INFOPATH に追加する必要があります。ただし、シンボリックリンクの作成先ディレクトリについて書き込み権限が必要です。このオプションは一般的に知られた /usr/local/bin などのディレクトリ（ただし TEX 関連コマンドがまだインストールされていない場所）に TEX 関連コマンドを配置するためのものです。シンボリックリンクの作成先にシステムディレクトリを指定するなどして、元々システムに存在していたファイルを上書きしてしまわないように注意してください。安全のため、このオプションは有効にしないことをおすすめします。

after install, set CTAN as source for package updates（CTAN をインストール後のパッケージ更新元に設定） TEX Live を DVD または ISO からインストールする場合、このオプションはデフォルトで有効です。ほとんどの場合、TEX Live のユーザは年内に行われるパッケージのアップデートを受け取りたいと考えるためです。このオプションを無効化する場合があるとしたら、例えば DVD からは一部のインストールのみを行って、後に（別の方法で）補強したいような場合でしょう。いずれにせよ、インストール時およびインストール後に利用するパッケージリポジトリは、必要に応じて独立に設定することが可能です。詳細は 3.3.1 節と 3.4.3 節を参照してください。

Windows のみで利用可能なオプションも、上級者向けの GUI インターフェースで利用可能です：

adjust searchpath（検索パスを調整） このオプションによってすべてのプログラムが確実に検索パス中で TEX Live のバイナリを見つけられるようにします。

add menu shortcuts（メニューにショートカットを追加） このオプションが有効の場合、スタートメニューに TEX Live サブメニューが追加されます。ここには ‘TeX Live

menu' と 'No shortcuts' の他に 'Launcher entry' という第三の選択肢もあります。このオプションについては 4.1 節で説明します。

file associations（拡張子の関連付け） このオプションでの選択肢は 'Only new'（拡張子の関連付けを行うが、既に存在する関連付けの上書きは行わない）、'All'、'None' の 3 つです。

install TEXworks front end（統合開発環境 TEXworks をインストール）

すべての設定が済みましたら、テキストモードでは 'I' をタイプ、GUI モードでは 'Install' ボタンを押すことによってインストールを実行してください。インストールが完了したら（他にすることがなければ）3.4 節までスキップしてください。

3.3 install-tl のコマンドラインオプション

次を実行すると、install-tl のコマンドラインオプションの一覧を確認することができます：

```
> install-tl -help
```

オプション名を指定するには - または -- を利用できます。以下では、一般的なオプションについてのみ説明します：

-gui インストーラを（もし利用可能なら）GUI モードで起動します。GUI モードの実行にはバージョン 8.5 以上の Tcl/Tk が必要です。古い macOS には同梱されていましたが、Big Sur より後のバージョンの場合は MacTEX インストーラを使用しないのであれば、別途自力でインストールする必要があります。Windows 用の TEX Live には Tcl/Tk が同梱されているため特に何もする必要はありません。古いオプションである -gui=perltk と -gui=wizard は現在も利用可能ですが、普通の GUI インターフェースを起動します。Tcl/Tk が利用可能でない場合、このオプションが指定された場合でもインストーラはテキストモードで続行されます。

-no-gui インストーラを必ずテキストモードで起動します。

-lang ⟨言語コード⟩ 指定した ⟨言語コード⟩（通常 2 文字）にあたる言語を使用します[*7]。インストーラは自動的に最適な言語で起動しますが、もし自動判定に失敗した場合や最適な言語が利用不可能な場合は、英語を使用します。インストーラで利用可能

[*7] 訳注：日本語を使用したい場合は -lang ja とします。

な言語の一覧は install-tl -help で確認できます。

-portable TEX Live を USB メモリなどにポータブルインストールします。ポータブルインストールは、インストーラの起動後にテキストモードの V コマンドまたは GUI モードのメニューを利用して選択することもできます。詳細は 4.2 節を参照してください。

-profile ⟨ファイル⟩ インストールプロフィール ⟨ファイル⟩ を読み込み、非対話的にインストールを実行します。なおインストーラは常に texlive.profile を tlpkg ディレクトリに書き込みます。このファイルを利用すると、まったく同じ設定でインストールを（例えば別のシステムでも）再実行することが可能です。別の活用方法として、ご自身で用意したカスタムプロフィール（生成された texlive.profile を改変するのが簡単です）を使用したり、空のファイルを指定してすべてをデフォルト設定でインストールしたりすることもできます。

-repository ⟨リポジトリ⟩ パッケージリポジトリ（ディレクトリまたは URL）を指定します。下の 3.3.1 節を参照してください。

-in-place（警告：このオプションは何をしようとしているのか完全に承知している場合以外は使用しないでください。）すでに rsync や svn またはその他の手段で TEX Live のコピー（その入手元については https://tug.org/texlive/acquire-mirror.html を参照）が手許にある場合、このオプションを利用するとそのコピーをそのまま利用し、追加で必要となるインストール後の処理だけを実行します。なお tlpkg/texlive.tlpdb は上書きされるので十分注意してください。また、既にあるパッケージの除去は手動で行う必要があります。このオプションは起動後にインストーラのインターフェースで有効にすることはできません。

3.3.1 -repository オプション

デフォルトのネットワーク上パッケージリポジトリは https://mirror.ctan.org によって自動的に選択される CTAN ミラーです。

このデフォルトを上書きしたい場合は、使用したいリポジトリを ftp:, http:, https:, file:/ のいずれかからはじまる URL、または通常のディレクトリパスによって指定します（http:, https:, ftp: から始まるリポジトリ指定を行った場合、末尾の '/' や '/tlpkg' は無視されます）。

例えば CTAN の特定のミラーサーバを http://ctan.example.org/tex-archive/systems/texlive/tlnet/ のように指定した場合、実際のホスト名と、そのホストにお

ける CTAN のトップレベルパスを ctan.example.org/tex-archive の代わりに使用します。CTAN ミラーの一覧は https://ctan.org/mirrors で確認できます。

もし与えられた引数が（パスまたは file:/ URL の形で）ローカルのディレクトリを指していた場合、その下にある archive サブディレクトリ内の圧縮ファイルが（仮に展開済みのものがあったとしても）使用されます。

3.4 インストール後のアクション

場合によっては、インストール後に多少の手順を行う必要があるかもしれません。

3.4.1 Unix 系における環境変数の設定

インストール時に標準ディレクトリにシンボリックリンクを作成している（3.2.4 節を参照）場合は、環境変数を変更する必要はありません。それ以外の場合、Unix 系システムでは各自適切なプラットフォーム向けのバイナリのあるディレクトリを検索パスに追加する必要があります（Windows ではインストーラが行うので気にする必要はありません）。

TEX Live にサポートされている各プラットフォームには TEXDIR/bin 以下に専用のサブディレクトリがあります。専用サブディレクトリのリストと、それぞれに対応するプラットフォームについては図 5 を参照してください。

さらに、必要であれば man ページと info マニュアルを含むディレクトリを、それぞれの検索パスに追加することができます。ただし man ページについては PATH を設定することによって自動的に検索可能になる場合もあります。

例えば、Intel x86 アーキテクチャ上の GNU/Linux で bash など Bourne シェルと互換性のあるシェルを使用していて、なおかつ TEX Live デフォルトのディレクトリ設定の環境の下では $HOME/.profile（もしくは .profile から読み込まれている別のファイルでも構いません）に次のような行を追加することで環境変数を設定できます：

```
PATH=/usr/local/texlive/2024/bin/x86_64-linux:$PATH; export PATH
MANPATH=/usr/local/texlive/2024/texmf-dist/doc/man:$MANPATH; export MANPATH
INFOPATH=/usr/local/texlive/2024/texmf-dist/doc/info:$INFOPATH; export INFOPATH
```

csh や tcsh の場合は、編集すべきファイルは典型的には $HOME/.cshrc で、次のような行を追加します：

```
setenv PATH /usr/local/texlive/2024/bin/x86_64-linux:$PATH
setenv MANPATH /usr/local/texlive/2024/texmf-dist/doc/man:$MANPATH
setenv INFOPATH /usr/local/texlive/2024/texmf-dist/doc/info:$INFOPATH
```

利用しているプラットフォームが x86_64-linux でない場合は、適切なプラットフォーム名を使用してください。またデフォルトのディレクトリ以外の場所にインストールを行った場合は、ディレクトリ名も対応するものに変更してください。TEX Live インストーラはインストールの完了時にそれぞれのフルパスを報告します。

すでに PATH をどこかの起動時設定ファイルで設定している場合は、適当だと思う方法で TEX Live ディレクトリを含めるようにしてください。

3.4.2 環境変数をグローバルに設定する

もし前節で説明したような環境変数の変更をシステム全体に適用したい場合や、新しいユーザにも適用したい場合は、自力で設定を行ってください。なぜなら、そのようなことを実現する方法は、システムによってあまりに異なるからです。

ここでは 2 つほど大まかな方針を示しておきます。第一に /etc/manpath.config というファイルの存在を確かめて、もし存在していたら次のような内容を追記します：

```
MANPATH_MAP /usr/local/texlive/2024/bin/x86_64-linux \
            /usr/local/texlive/2024/texmf-dist/doc/man
```

第二に各種の検索パスや環境変数のデフォルト値を設定するためのファイル /etc/environment の内容を確認することです。

各 Unix 系システム向けのバイナリディレクトリには、man という名前の texmf-dist/doc/man を指すシンボリックリンクが作成されています。一部の man プログラム（例えば macOS の標準 man コマンド）は、これを利用して自動的に man ページの設定を最適化してくれる場合があります。

3.4.3 DVD インストール後のインターネットを利用したアップデート

TEX Live を DVD からインストールした後、インターネットを利用してアップデートを行いたい場合は、以下のコマンドを検索パスの設定の後に行ってください：

```
> tlmgr option repository https://mirror.ctan.org/systems/texlive/tlnet
```

これによって以降 tlmgr は最寄りの CTAN ミラーを利用してアップデートを行うようになります。DVD からのインストール時に 3.2.4 節で説明したオプションを介してこの設定をデフォルトにしておくこともできます。

もしミラーの自動選択では問題がある場合は、https://ctan.org/mirrors から特定の CTAN ミラーを指定することもできます。その際は tlnet サブディレクトリまでのパ

スを正確に指定するようにしてください。

3.4.4 XƎTEX と LuaTEX のためのシステム設定

XƎTEX と LuaTEX は TEX ツリーにあるものだけでなく、システムにインストールされているすべてのフォントを使用することができます。TEX Live 外のシステムフォントは通常フォント名（例えば `Liberation Serif`）を利用してアクセスされますが、システムにおけるファイル名も使用することができます。

関連する話題として、TEX Live ディストリビューションが提供するフォントをシステムフォントとして利用可能にする（すなわちフォント名でアクセスできるようにする）にはどうすればよいかというものがあります。

■LuaTEX の場合 特に何もしなくてもフォント名でアクセスすることが可能です。TEX Live に含まれるすべてのフォントは、フォント名でもファイル名でも同じようにアクセス可能であるはずです。これは LATEX と plain TEX の両方をサポートする luaotfload パッケージのおかげです。なお luaotfload のフォント名インデックスは新しいフォントについては再生成される必要がありますが、これは未知のフォントを読み込もうとすると自動的に実行されます。

■XƎTEX の場合 Windows では TEX Live のフォントも自動的に利用できる状態になっているはずです[*8]。macOS については、別の文献を参照してください[*9]。macOS 以外の Unix 系システムの場合の手順は以下の通りです。

xetex パッケージがインストールされている場合[*10]、必要な設定ファイルが `TEXMFSYSVAR/fonts/conf/texlive-fontconfig.conf` に作成されています。TEX Live のフォントをシステムフォントとして利用可能にするためには以下を実施します：

1. その `texlive-fontconfig.conf` ファイルを（典型的には）`/etc/fonts/conf.d/09-texlive.conf` にコピー

---

*8 Windows 向け TEX Live に含まれる fc-cache プログラムを実行することで実現されています。

*9 訳注：TEX Live 内のフォントを個別にインストールするだけであれば、当該フォントの実体ファイル（例えば ‘`kpsewhich` ⟨フォントファイル名⟩’ を実行すれば見つけられます）を Finder 上で表示してダブルクリックすることで Font Book アプリが起動し、インストールすることが可能です。TEX Live に含まれるインストール可能なフォントを一括で処理する場合には `TEXMFDIST/fonts/` 以下にあるフォントファイルを `/Library/font`（システム全体にインストールする場合）や `~/Library/font`（特定ユーザのみにインストールする場合）にすべてコピーすればよいようです。

*10 最初からインストールされている場合も、後から追加された場合も同様です。

2. fc-cache -fsv を実行

上記を実行するのに十分な権限がない場合や、1 ユーザのみを対象に設定を適用したい場合には以下のようにします：

1. その texlive-fontconfig.conf ファイルを（典型的には）~/.fonts.conf.d/09-texlive.conf（~ はホームディレクトリを表す）にコピー
2. fc-cache -fv を実行

fc-list を実行するとシステムにあるフォントの一覧を確認することができます。また fc-list : family style file spacing（タイプミスに注意してください）とすると、もう少し詳細な情報を表示することができます。

3.4.5 ConTEXt LMTX と MKIV

「古い」ConTEXt（Mark IV と MkIV）と「新しい」ConTEXt（LMTX と MkXL）はいずれも TEX Live のインストール後はそのままで動く状態になっているはずで、tlmgr を使用してアップデートを行っている限りは特に特別な設定を行う必要はありません。

ConTEXt は Kpathsea ライブラリやシステムワイドの ls-R ファイルを使用していません。代わりに、ConTEXt はユーザごとのファイル名データベースをすべての既知のツリーについて管理しています。mktexlsr を実行してもこれらのデータベースは直接は更新されませんが、ConTEXt は次に実行された際に自動的にこれらのデータベースを再生成します[*11]。

もし何らかの理由でこれらのデータベースを手動で再生成したい場合には、以下のいずれかもしくは両方を実行します。

```
> mtxrun --generate           # LMTX
> mtxrun --luatex --generate  # MkIV
```

なお、この操作によって生成されるファイルは TEXMFCACHE（デフォルト値は TEXMFSYSVAR;TEXMFVAR）以下に保存されます。

ConTEXt は TEXMFCACHE に列挙されているパスをすべて読み込み、そして書込み可能な最初の場所にキャッシュを作成します。読み込みの際は、複数のキャッシュが見つかった場合は（見つかったキャッシュの中で）一番最後のパスにあるものが優先されます。

[*11] TEX Live 専用の cont-sys.mkiv と cont-sys.mkxl がその実装です。

より詳細な情報については https://wiki.contextgarden.net/LMTX および https://wiki.contextgarden.net/Running_Mark_IV を参照してください。

3.4.6 ローカルおよび個人用のマクロを利用する

ローカルおよび個人用のマクロの配置方法については 2.3 節でも簡単に言及しましたが、TEXMFLOCAL（デフォルトでは /usr/local/texlive/texmf-local または %SystemDrive%\texlive\texmf-local）はシステム全体に適用したいローカルなフォントやマクロを、TEXMFHOME（デフォルトでは $HOME/texmf または %USERPROFILE%\texmf）はユーザごとの個人用フォントやマクロを配置するためのものです。これらのディレクトリは毎年のリリースとは切り離して管理されることが意図されており、そこに含まれているファイルは自動的に各 TEX Live リリースから検索できるようになっています。したがって、TEXMFLOCAL の値をメインの TEX Live ツリーとあまりかけ離れた位置に変更することは避けるべきです。さもないと将来のリリースの際に手動で変更を行う必要が発生する可能性があります。

いずれのツリーについても、それぞれのファイルは TDS に基づいたサブディレクトリに配置されているべきです（https://tug.org/tds や texmf-dist/web2c/texmf.cnf を参照してください）。例えば LATEX のクラスファイルやパッケージは TEXMFLOCAL/tex/latex や TEXMFHOME/tex/latex、あるいはそれより下の階層に配置されているべきです。

TEXMFLOCAL については最新の状況を反映したファイル名データベースがないと、その中にあるファイルが見つけられません。データベースは mktexlsr コマンドを実行するか、GUI 版 TEX Live マネージャで ‘Update filename database’ ボタンを押すことによって更新することができます。

デフォルトでは、これまで見てきたように各 TEXMF ツリーを表す変数は 1 つのディレクトリを指すように定義されています。しかし、これは絶対そうでなければならないわけではありません。例えば、もし複数のバージョンの大きなパッケージを簡単に切り替える必要があるような場合は、複数のツリーを用意した上で TEXMFHOME をブレースで囲われたカンマ区切りのディレクトリ名リストに設定するとよいでしょう：

```
TEXMFHOME = {/my/dir1,/mydir2,/a/third/dir}
```

ブレース展開については 7.1.5 節により詳しい解説があります。

3.4.7 サードパーティフォントを利用する

残念ながら TEX と pdfTEX に関しては、これは泥沼への入り口です。TEX インストールの深淵に飛び込みたい場合を除いては、すべての TEX 処理系でサードパーティフォントを利用できるようにすることは諦めた方が無難です。TEX Live には数多くのフォントが含まれているので、それらの中に適当なものがないか探してみてください。TUG のウェブページ https://tug.org/FontCatalogue には主要な TEX ディストリビューションで利用可能なフォントのほとんどすべてが、さまざまな観点から分類されて、一覧になっています。

もしどうしてもサードパーティフォントを一般の TEX 処理系で利用できるように設定を行う必要がある場合は、https://tug.org/fonts/fontinstall.html にその方法について（可能な範囲での）解説がありますので、参考にしてください。

また XETEX または LuaTEX（2.4 節）の利用も検討に値します。これらの処理系であれば TEX のために特別な設定を行わなくても OS にインストールされたあらゆるフォントを利用することができます（ただし、システムフォントを利用すると、その文書は他の環境では簡単に作成できなくなる点には注意してください）。

日本語 TEX ユーザ向けノート

上述の通り、XETEX や LuaTEX を除く古典的な TEX 処理系でサードパーティフォントを利用するのは難しいです。pTEX や upTEX 用にサードパーティの和文フォントを設定するのもやはり難易度が高く、初心者にはおすすめできません（自由にフォントを選びたい場合は LuaTEX-ja の利用を検討してください）。しかし、幸い TEX Live にはよく使われる和文フォントの埋め込みについて設定を自動化するためのプログラム `kanji-config-updmap` が付属しています。詳細は TEX Live に含まれる `kanji-config-updmap` のドキュメント（日本語）を参照してください。

3.5 インストールした TEX Live をテストする

TEX Live をインストールした後、美しい文書やフォントの作成を始めるために TEX Live が正常に動作するかテストしたいと思うのが普通でしょう。

おそらく、多くの人がまず最初に動作確認したいのはファイル編集を行うフロントエンドでしょう。TEX Live は Windows（のみ）では TEXworks（https://tug.org/

texworks）を、MacTEX は TEXShop（https://pages.uoregon.edu/koch/texshop）をインストールします。他の Unix 系システムではどのエディタを利用するかはユーザが決めてください。エディタの選択肢はとてもたくさんあり、その一部は次のセクションでリストアップされています。また https://tug.org/interest.html#editors も参考になるかと思います。基本的にはどんなテキストエディタでも使用できます。TEX のための特別な機能は、なくてならないものではありません。

残りのセクションでは、新しい TEX Live が正常に動作するか確認する基本的な手順を解説します。なお、ここでは Unix コマンドを使用した方法を紹介します。macOS や Windows のユーザは、GUI を利用してテストをしたいと考えるかもしれませんが、根本的な部分は変わりません。

1. まずはじめに、tex プログラムが実行可能であることを確認しましょう：

   ```
   > tex -version
   TeX 3.14159265 (TeX Live ...)
   Copyright ... D.E. Knuth.
   ...
   ```

   もしこれを実行したとき、正しいバージョン情報・コピーライト情報の代わりに ‘command not found’ エラーが出たり、古いバージョン情報が表示される場合は、正しい bin ディレクトリが PATH に追加されていない可能性があります。そのような場合は p. 30 にある環境変数設定の情報を確認してください。

2. 基本的な LATEX ファイルを処理して、PDF を生成してみましょう：

   ```
   > pdflatex sample2e.tex
   This is pdfTeX 3.14...
   ...
   Output written on sample2e.pdf (3 pages, 142120 bytes).
   Transcript written on sample2e.log.
   ```

   sample2e.tex やその他のファイルが見つからないというエラーが出る場合は、古い環境変数の設定や設定ファイルの影響を受けてしまっている可能性が高いです。まずは TEX 関係の環境変数をすべてクリアすることをおすすめします。（より詳細に分析したい場合は、TEX 自身に何を検索し、何を見つけたのか報告させるのが良いでしょう。詳細は 7.2.4 節を参照してください。）

3. 処理結果の PDF ファイルをプレビューしてみます。例えば次のようにします：

```
> xpdf sample2e.pdf
```

すると、新しいウィンドウがポップして LATEX の基礎について説明した素敵な文書が表示されるはずです（ところで、もし TEX を使うのが初めてでしたら、その内容も一読の価値があります）。
もちろん PDF ビューアは他にも色々なものがあります。Unix 系システムでは `evince` や `okular` がよく用いられます。Windows 向けのものとしては Sumatra PDF（https://www.sumatrapdfreader.org/free-pdf-reader.html）をおすすめします。TEX Live には PDF ビューアは含まれていないので、PDF の閲覧には何かしらを必ず別途インストールする必要があります。

4. もちろん、いまでも TEX の元来の出力フォーマットである DVI を生成することもできます：

```
> latex sample2e.tex
```

5. DVI をプレビューするには：

```
> xdvi sample2e.dvi    # Unix
> dviout sample2e.dvi  # Windows
```

なお `xdvi` コマンドを利用するには X ウィンドウシステムが正しく動作している必要があります。正常に動作していない場合や、環境変数 `DISPLAY` の設定がまずい場合は '`Can't open display`' エラーが発生します。

6. 続いて DVI から PostScript ファイルを作成してみましょう：

```
> dvips sample2e.dvi -o sample2e.ps
```

7. あるいは DVI から PDF に変換することもできます。これは pdfTEX（あるいは XETEX や LuaTEX）を使わない手段として有用な場合があります[*12]。

```
> dvipdfmx sample2e.dvi -o sample2e.pdf
```

8. `sample2e.tex` 以外でテストのために利用できる標準的なファイルを以下に示して

[*12] 訳注：日本語組版では今でもよく用いられる pTEX 系の処理系は DVI を出力するため、現在も `dvipdfmx` を用いて PDF 生成を行う手順は一般的です。

おきます：

`small2e.tex` `sample2e.tex` よりもシンプルな文書です。なにか問題が発生したときに、より入力サイズの小さいものとして利用できます。

`testpage.tex` プリンタにオフセットがあるかどうかを確認するための文書です。

`nfssfont.tex` フォントテーブルを出力・テストするためのものです。

`testfont.tex` plain TEX を用いて、同じくフォントテーブルを出力するものです。

`story.tex` 最も標準的な (plain) TEX のテスト用ファイルです。`tex story.tex` を実行後、`*` プロンプトに '`\bye`' を入力する必要があります。

9. xetex パッケージをインストールしている場合には、XƎTEX がシステムフォントを正常に利用できるかどうかテストすることができます：

```
> xetex opentype-info.tex
This is XeTeX, Version 3.14...
...
Output written on opentype-info.pdf (1 page).
Transcript written on opentype-info.log.
```

もし "Invalid fontname 'Latin Modern Roman/ICU'..." というエラーメッセージが表示された場合は、TEX Live に含まれるフォントを見つけられるように設定を行う必要があります。3.4.4 節を参照してください。

3.6 アンインストール

TEX Live をアンインストールには、以下のコマンドを実行します（Windows については後述）：

```
> tlmgr uninstall --all
```

上記を実行すると、本当にアンインストールを行うかどうか確認されます。この確認を行わない限りは、アンインストールは実行されません。（なお `--all` の付かない `uninstall` アクションは、1 つ 1 つのパッケージを個別にアンインストールするためのものです。）

このアクションでは、以下に列挙するユーザ固有のディレクトリは削除しません（2.3 節も参照）。

TEXMFCONFIG ユーザによる設定変更を行うためのツリーです。変更済みの設定を維持したい場合は、削除する前に復元方法を確認するようにしてください。

TEXMFVAR 自動生成された実行時データ（ローカルのフォーマットファイルなど）を保存することを意図したツリーです。他の目的で利用していない限りは削除しても問題はありません。

TEXMFHOME このツリーにはユーザ自身がインストールしたファイル（ディストリビューションに未収録のものなど）だけが含まれます。完全に TEX の利用をやめる場合や、まったく新しい TEX 環境を新たに構築したい場合を除いては、このツリーは削除しない方がよいでしょう。

こうしたツリーのパス位置は kpsewhich -var-value=⟨ツリー⟩ を実行することで確認することができます。

tlmgr によるアンインストールではインストール後のアクション結果を復帰するところまでは行いません。例えばシェルの初期化ファイルにおける PATH の変更や TEX Live 内のフォントをシステムに認識させる設定はそのままになります（3.4 節参照）。必要な場合は、これらの設定は自力で元に戻す必要があります。

Windows では GUI を用いてアンインストールを行うことができます。6.1 節を参照してください。

3.7 その他のダウンロード可能なソフトウェア

TEX 初心者や LATEX 文書を実際に作成するにあたってヘルプが必要な方は、ぜひ https://tug.org/begin.html にアクセスしてみてください。いくつかの初心者向けの情報を掲載しています。

ここではインストールの検討に値する各種ツールへのリンクを列挙しておきます：

Ghostscript https://ghostscript.com/（PostScript/PDF インタプリタ）

Perl https://www.perl.org/（追加パッケージは CPAN（https://www.cpan.org/）から入手可）

ImageMagick https://imagemagick.org/（画像の加工や変換）

NetPBM http://netpbm.sourceforge.net/（こちらも画像用ツール）

TEX 向きのエディタ テキストエディタの選択肢はかなり多岐にわたり、その選択は完全に各自の好みによります。ここでは主なものをアルファベット順に紹介しておきます（一部は Windows 専用です）。

- GNU Emacs: https://www.gnu.org/software/emacs/emacs.html（多くの主要なプラットフォームで利用可能）
- AucTEX: https://www.gnu.org/software/auctex（Emacs プラグイン。Emacs のパッケージマネージャ ELPA でインストール可能。ソースは CTAN からも入手可）
- SciTE: https://www.scintilla.org/SciTE.html
- Texmaker: https://www.xm1math.net/texmaker
- TEXstudio: https://texstudio.org/（Texmaker の拡張プロジェクト）
- TEXnicCenter: http://www.texniccenter.org
- TEXworks: https://tug.org/texworks（Windows 版 TEX Live では標準添付）
- Vim: https://www.vim.org
- WinEdt: https://tug.org/winedt, http://www.winedt.com（シェアウェア）
- WinShell: https://www.winshell.de

4 特殊なインストール

ここまでは基本的なインストール手順について説明してきました。このセクションでは、いくつかの特殊なインストールについて解説します。

4.1 共有インストール

TEX Live はネットワーク上の複数のシステムで共有できるように設計されています。標準のディレクトリ構成では、いかなるパスもハードコードされていません。TEX Live に含まれるプログラムは必要なファイルを相対パスによって探索します。このことは、TEX Live の根幹にあたる設定ファイル `$TEXMFDIST/web2c/texmf.cnf` を見ればわかります。そこには

```
TEXMFROOT = $SELFAUTOPARENT
...
TEXMFDIST = $TEXMFROOT/texmf-dist
...
TEXMFLOCAL = $SELFAUTOGRANDPARENT/texmf-local
```

のような記述が並んでいるはずです。この仕組みのおかげで、各プラットフォーム向けの実行ファイルを、それぞれの検索パスに追加するだけで TEX Live が機能するように設定できます。

同様に、一度ローカルに TEX Live をインストールし、そのすべての階層をネットワーク上の場所に移動することも可能です。

Windows 向けには、TEX Live は `tlaunch` というランチャーを含んでいます。そのメイン画面のメニューには種々の TEX 関連プログラムおよびドキュメントについてアクションを起こすためのボタンがあります。この画面は `ini` ファイルによってカスタマイズすることも可能です。初回起動時には、それらのボタンは通常の Windows 向けポストインストールを実行（すなわち検索パスと拡張子の関連付けを更新）しますが、この操作は現在ログイン中のユーザにしか適用されません。したがって、ローカルネットワークにアクセスできる各システムで TEX Live を利用できるようにするためには、`tlaunch` にメニューショートカットを追加するだけで事足ります。詳細については `tlaunch` のマニュアルを参照してください（`texdoc tlaunch` を実行するか https://ctan.org/pkg/tlaunch にアクセスすると閲覧できます）。

4.2 ポータブル（USB）インストール

TEX Live インストーラに `-portable` オプションを与えると（テキストモードで `V` コマンドを実行したり GUI モードで同等のオプションを利用した場合も同様です）TEX Live は完全に自己完備のインストールを標準的なルート以下に作成します。これはシステムのいかなる環境にも依存しません。このような構成の TEX Live は USB メモリに直接インストールしたり、後から USB にコピーしたりすることができます。

具体的には、ポータブルインストールでは `TEXMFHOME`, `TEXMFVAR`, `TEXMFCONFIG` のデフォルト値をそれぞれ `TEXMFLOCAL`, `TEXMFSYSVAR`, `TEXMFSYSCONFIG` と一致するように設定することで、自己完結するようになっています。そのため、ユーザごとの設定やキャッシュが作られることはありません。

ポータブルインストールされた TEX を実行するには、通常通り実行ファイルのある適切なディレクトリを（端末セッションの間）検索パスに追加する必要があります。

Windows では、ポータブルインストールのルートにある `tl-tray-menu` をダブルクリックすることによって一時的な ‘tray menu’ が作成され、これを用いていくつかの一般的なタスクを実行することができます：

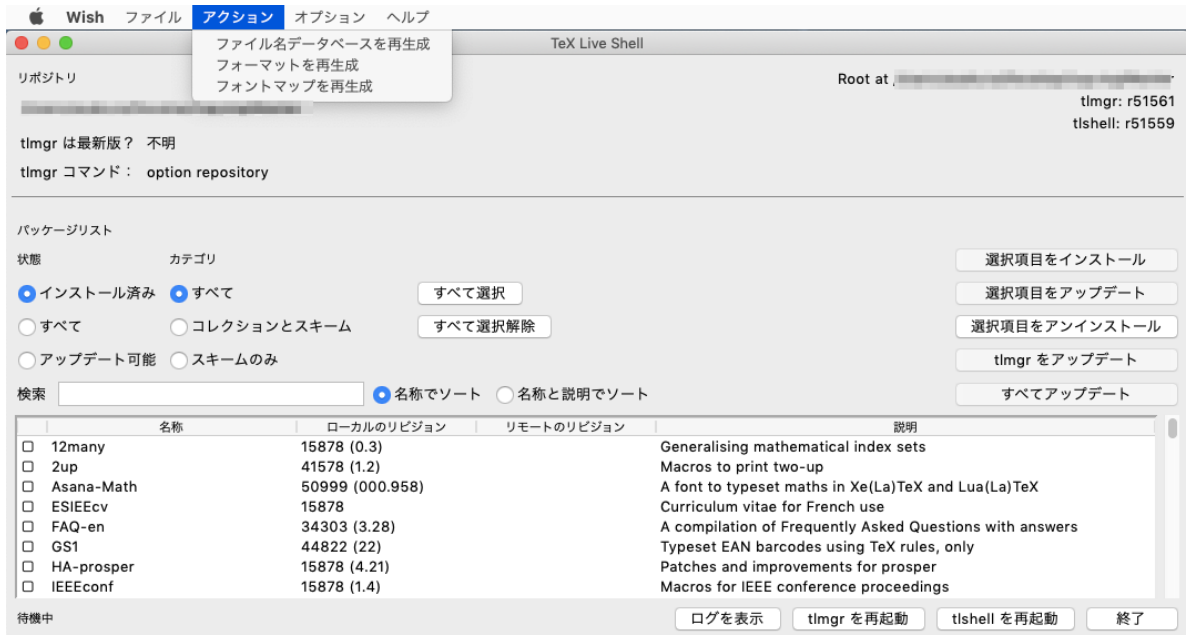
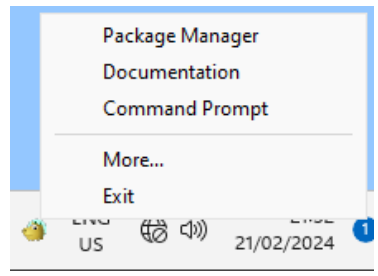


図 9 tlshell GUI のアクションメニュー（macOS）



この画面で ‘More. . . ’ を押すと、このメニューのカスタマイズ方法についての説明が表示されます。

5 tlmgr: TEX Live マネージャ

TEX Live にはインストール後の TEX Live 管理のために tlmgr というプログラムが同梱されています。その主な機能は：

- 個別パッケージのインストール、アップデート、バックアップ、アンインストール（依存関係管理も可）
- パッケージの検索と詳細の表示
- プラットフォームの表示、追加、削除
- インストールオプションの変更（デフォルトの用紙サイズやソースの場所）

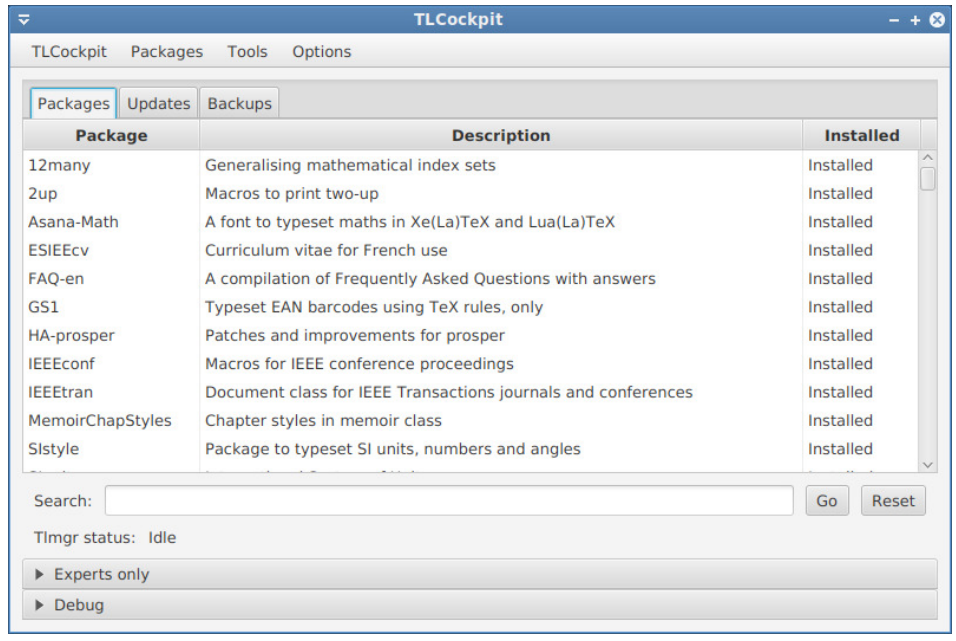


図 10 tlmgr の新しい GUI（tlcockpit）

texconfig の機能はすべて tlmgr にも備わっています。TEX Live チームは texconfig の操作に慣れている方々のために依然としてその配布およびメンテナンスを継続していますが、tlmgr を使用することをおすすめします。

5.1 tlmgr の GUI

TEX Live には tlmgr の GUI フロントエンドが複数含まれています。重要なものが 2 つあります。ひとつは図 9 で示されている tlshell という Tcl/Tk を利用した GUI で、Windows で利用可能です。もうひとつは図 10 で示されている tlcockpit で、動作にはバージョン 8 以降の Java と JavaFX が必要です。これらはいずれも独立したパッケージです。

tlmgr もネイティブの GUI モードを持っています（図 11）：

```
> tlmgr -gui
```

しかし、この GUI 拡張の実行には Perl/Tk が必要です。現在は Windows 向けの TEX Live には Perl/Tk モジュールは同梱されていないので注意してください。

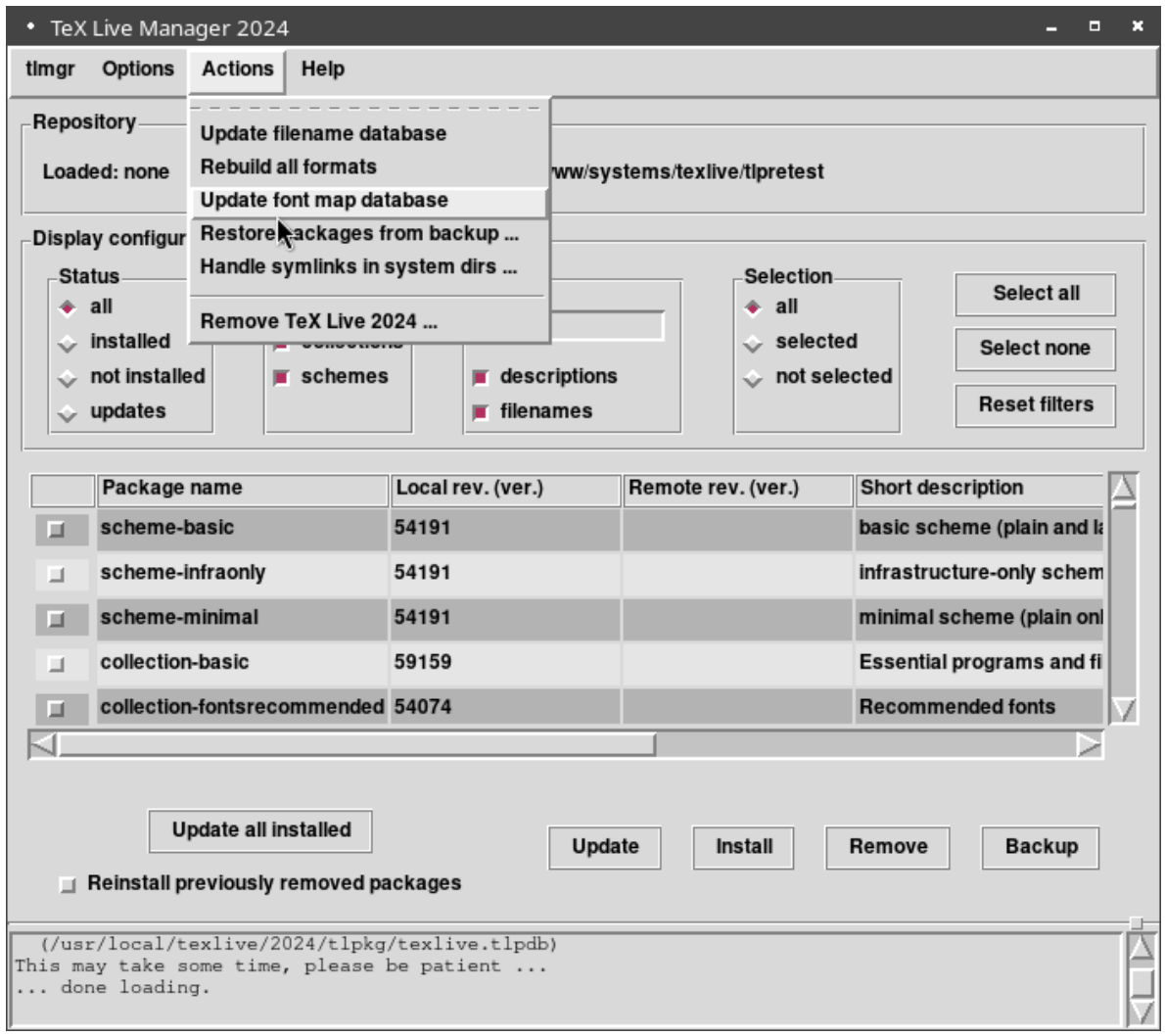


図 11 tlmgr のレガシー GUI モード。‘Load’ 後のメインウィンドウ

5.2 コマンド使用例

最初のインストールののち、TEX Live を最新の状態にアップデートするには：

```
> tlmgr update -all
```

もし実際のアップデートを実行する前に何が起きるかを確認したい場合は：

```
> tlmgr update -all -dry-run
```

または（出力メッセージを減らすなら）

```
> tlmgr update -list
```

もう少し複雑な例として、XƎTEX 用のコレクションをローカルディレクトリから追加する方法を示します：

```
> tlmgr -repository /local/mirror/tlnet install collection-xetex
```

これを実行すると、次のような出力が表示されます（一部省略しています）：

```
install: collection-xetex
install: arabxetex
...
install: xetex
install: xetexconfig
install: xetex.i386-linux
running post install action for xetex
install: xetex-def
...
running mktexlsr
mktexlsr: Updating /usr/local/texlive/2024/texmf-dist/ls-R...
...
running fmtutil-sys --missing
...
Transcript written on xelatex.log.
fmtutil: /usr/local/texlive/2024/texmf-var/web2c/xetex/xelatex.fmt installed.
```

上の出力からわかるように、tlmgr は依存パッケージもインストールし、ファイル名データベースの更新やフォーマットの（再）生成など必要なインストール後処理も自動的に実行しています。ここでは XƎTEX 用の新しいフォーマットが生成されています。

パッケージ（またはコレクション、スキーム）の詳細を確認するには次のようにします：

```
> tlmgr show collection-latexextra
```

これにより以下のような出力が得られるはずです：

```
package:    collection-latexextra
category:   Collection
shortdesc:  LaTeX supplementary packages
longdesc:   A very large collection of add-on packages for LaTeX.
installed:  Yes
revision:   46963
sizes:      657941k
```

最後に、最も大事なことですが、フルドキュメントを確認するには https://tug.org/

texlive/tlmgr.html にアクセスするか、以下を実行してください：

```
> tlmgr -help
```

6 Windows 向けの情報

6.1 Windows 専用の機能

Windows では TEX Live インストーラはいくつか追加の処理を行います：

メニューとショートカット スタートメニューに 'TEX Live' サブメニューを追加します。このサブメニューから tlshell（tlmgr の GUI）や dviout などの GUI プログラムや、いくつかのドキュメントにアクセスすることができます。

拡張子の関連付け 設定が有効になっている場合、TEXworks, Dviout などを、それぞれが扱う拡張子に関連付けます。あるいは、関連するファイルを右クリックして「プログラムから開く」を選択した場合にこれらのプログラムが候補に挙がるようになります。しかしながら、より優先度の高い「ユーザ選択」のファイル関連付け（対話的にのみ設定できる）が存在する場合、そちらが優先されることになります。

PostScript サポート PostScript ファイルは現在では PSviewer によって一時的に PDF に変換された上でデフォルトの PDF ビューアで表示されるようになりました。各種のビットマップ画像を右クリックして「プログラムから開く」を選択した場合に bitmap2eps を選択できるようになります。bitmap2eps はシンプルなスクリプトで、バックエンドとして sam2p や bmeps を利用します。

自動パス設定 Windows では手動でパス設定を行う必要はありません。

アンインストーラ TEX Live のインストーラは「プログラムの追加と削除」メニューの中（管理者権限でインストールした場合）または TEX Live メニューの中（単一のユーザのみにインストールした場合）に TEX Live をアンインストールするための項目を追加します。

書き込み禁止 管理者として TEX Live をインストールした場合、TEX Live のディレクトリは書き込み禁止に設定されます（通常の NTFS フォーマット内蔵ディスクにインストールした場合）。

また tlaunch（4.1 節）も参照してみてください。

6.2 Windows 向けに追加されているソフトウェア

TeX Live を完全にインストールするためには、Windows マシンには通常存在していないプログラムが必要となります。そのため TeX Live は Windows 向けに一部の依存パッケージを同梱しています。以下に列挙するプログラムは Windows 版の TeX Live にのみ含まれています：

Perl, Tcl/Tk, Ghostscript Perl と Ghostscript は重要なプログラムであること、また TeX Live インストーラの GUI モードや tlshell が Tcl/Tk を利用することから TeX Live には隠されたそれらのコピーが含まれています。これらのプログラムは TeX Live のプログラムからは利用可能ですが、環境変数やレジストリ設定の上では存在していないことになっています。TeX Live に同梱されている Perl と Tcl/Tk は完全版ではなく、また 3 プログラムともシステムに別途インストールされたものとは一切干渉しないはずです。TeX Live に含まれるインフラ以外のスクリプトを、外部の Perl 等で動作させたい場合の設定方法については 6.3 節を参照してください。

dviout DVI ビューア。初期状態では、ファイルを dviout でプレビューしようとすると、フォント生成が行われるはずです。これはスクリーンフォントがインストールされていないためです。しばらく使用していると、頻繁に利用されるフォントの生成が完了するため、フォント生成ウィンドウが現れることは稀になります。詳細はオンラインヘルプ（おすすめ）を参照してください。

TeXworks PDF ビューア機能をもつ TeX 用の統合開発環境。

各種コマンドラインツール TeX Live バイナリと同時に、多数の Windows 移植版の Unix コマンドもインストールされます。`gzip`, `zip`, `unzip` や poppler 関連の `pdfinfo`, `pdffonts` などです。

fc-list, fc-cache, ... これらのフォント管理ライブラリを利用して XeTeX から Windows 上のフォントを利用できます。fc-list を用いると XeTeX 拡張の `\font` コマンドに与えるべきフォント名を決定することができます。必要に応じて fc-cache を事前に実行してフォント情報を更新してください。

6.3 外部の Perl, Tcl/Tk, Ghostscript を利用する

通常 TeX Live は同梱の Perl, Tcl/Tk, Ghostscript を使用します。これはインフラだけでなく、TeX Live が収録するすべてのスクリプトについても同様です。ユーザ自身が別途インストールした外部の Perl 等を使用したい場合には、TeX Live ルート直下の `texmf.cnf` で設定を行うことができます。

外部の Perl を使用するには `texmf.cnf` に

```
TEXLIVE_WINDOWS_TRY_EXTERNAL_PERL = 1
```

という行を追加します。こうすることで、TeX Live（特に `bin/windows/runscript.tlu`）は TeX Live のインフラツール以外について検索パス内の `perl.exe` を探すようになります。これはここで紹介する 3 つの設定のうちで最も有益なケースが多いと考えられます。TeX Live の Perl も多数の追加モジュールをサポートしていますが、すべてのサードパーティスクリプトをサポートできるほどではありません。

Tcl/Tk の場合も同様で

```
TEXLIVE_WINDOWS_TRY_EXTERNAL_TCL = 1
```

という行を追加することで、TeX Live は検索パス内の `tclkit.exe`, `wish.exe`, `wish85.exe`, `wish86.exe`, `wish87.exe` を探すようになります。

Ghostscript の場合は設定方法が異なります。この場合、使用したい Ghostscript コマンドのファイル名もしくはフルパスを与える必要があります。

`TEXLIVE_WINDOWS_EXTERNAL_GS =` ⟨Ghostscript コマンドのパス⟩

もう 1 つの違いは、TeX Live 同梱の Ghostscript は完全なものであることです。ドキュメントとプリンタドライバのみが省略されています。したがって、ここに示した方法で TeX Live の設定を上書きする必要が生じることはほとんどないでしょう。

`texmf.cnf` については 7.1.2 節も参照してください。

6.4 ユーザプロファイルがホームディレクトリ扱い

Windows において、Unix 系のホームディレクトリに相当するのは `%USERPROFILE%` ディレクトリです。Windows Vista 以降の Windows では、そのデフォルト値は

C:\Users\<username> です。texmf.cnf 内および Kpathsea 全般において、~ は Windows と Unix 系のいずれにおいても適切に展開されます。

6.5 レジストリ

Windows ではほとんどすべての設定データはレジストリに保存されます。レジストリには数個のルートの下に階層的に構成されたキーが多数含まれています。簡単に言うと、インストールプログラムにとって最も重要なものは HKEY_CURRENT_USER と HKEY_LOCAL_MACHINE（略称は HKCU と HKLM）です。レジストリのキー HKCU はユーザのホームディレクトリ内を参照します（6.4 節を参照）。HKLM は通常 Windows ディレクトリ内のサブディレクトリです。

一部の項目については環境変数によってシステム情報を引き渡すことができますが、ショートカットの位置などはレジストリを変更しないことには設定を反映することができません。また環境変数の値を恒久的に変更するのにもレジストリを編集する必要があります。

6.6 パーミッション

最近のバージョンの Windows では、管理者のみに OS のほぼ全体へのアクセス権があり、一般ユーザにはありません。TEX Live は管理者権限がなくてもインストール可能なように工夫されています。

管理者権限でインストーラが実行されているときは、全ユーザ向けにインストールを行うためのオプションが利用可能です。このオプションを選択した場合、全ユーザ向けにショートカットが作成され、またシステムの検索パスも変更されます。逆に、このオプションを選択しなかった場合はシュートカットとメニュー項目はインストーラを実行したユーザ向けにのみ作成され、そのユーザの検索パスだけが更新されます。

管理者として実行するか否かに関わらず、インストーラは常にデフォルトの TEX Live ルートを %SystemDrive% 以下に設定します。インストーラは、必ず設定しようとしているルート位置が現在の実行権限で書き込み可能かどうか確認します。

検索パスに TEX が存在する状態で、一般ユーザとして TEX Live のインストールを実行しようとすると問題が発生します。システムの検索パスはユーザの検索パスよりも優先されるので、この状況で TEX Live をインストールしても優先的に利用されることはありません。フォールバックとして、インストーラは新しくインストールされた TEX Live バ

イナリのディレクトリがローカル検索パスの先頭にくるようコマンドプロンプトにショートカットを作成します。そのため、このフォールバックが有効なコマンドプロンプトからは常に新しい TEX Live が利用可能です。TEXworks がインストールされている場合、その検索パスについても TEX Live が先頭にくるよう設定されるため、同様に検索パスの問題は回避されるはずです。

Windows では管理者アカウントにログインしていても、明示的に管理者権限としての実行を指示する必要があることに注意してください。実際のところ、管理者アカウントにログインするだけではあまり意味がありません。肝心なのは、プログラムやショートカットを右クリックし「管理者として実行」を選択することです。

6.7 Windows と Cygwin でメモリ上限を増やす方法

Windows と Cygwin（Cygwin インストールの詳細については 3.1.4 節を参照してください）では、TEX Live に含まれるいくつかのプログラムがメモリを使い果たす場合があります。例えば asy で 2,500 万要素の実数配列を確保しようとしたり、LuaTEX で多書体組を行おうとしたりするとメモリが不足する可能性があります。

Cygwin ではユーザマニュアル（https://cygwin.com/cygwin-ug-net/setup-maxmem.html）の説明に従うことによってメモリ上限を増やすことができます。

Windows の場合はいくつかの手順を行う必要があります。まず moremem.reg というファイルに以下の内容を書き込みます：

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\Software\Cygwin]
"heap_chunk_in_mb"=dword:ffffff00
```

その上で、管理者権限で regedit /s moremem.reg を実行してください。（もしシステム全体ではなく、現在利用中のユーザのみについてメモリ上限を変更したい場合は HKEY_CURRENT_USER を使用してください。）

7 Web2C ユーザガイド

Web2C は TEX 関連プログラムのコレクションです。これにはオリジナルの TEX そのものに加えて METAFONT, METAPOST, BIBTEX などが含まれており、TEX Live

の心臓部と言えます。最新版のマニュアルやその他の情報はウェブサイト（https://tug.org/web2c）に掲載されています。

少し Web2C の歴史的経緯について触れておきます。Web2C のオリジナル作者は Tomas Rokicki で 1987 年に最初の TEX-to-C システムを開発しました。このシステムは Howard Trickey と Pavel Curtis による Unix 系システム下の changefile をベースにしていました。その後 Tim Morgan がこのシステムのメンテナとなり、その期間システムは Web-to-C と呼ばれていました。1990 年に Karl Berry が多くの協力者の助けを得ながらメンテナンスを引き継ぎました。1997 年に Olaf Weber がメンテナになりましたが、2006 年には再び Karl Berry の管理下に戻りました。

Web2C システムは Unix 系、Windows、macOS などの OS で動作します。このシステムでは Knuth による TEX のオリジナルソースと、文芸的プログラミングシステム WEB で書かれたいくつかの基本的なプログラムを使用し、それらを C 言語ソースコードに変換します。このような方法で管理されている TEX 関連のプログラムは以下の通りです：

bibtex　参考文献の管理
dvicopy　DVI ファイルで参照されている仮想フォントの展開
dvitomp　DVI を MPX（METAPOST 画像）に変換
dvitype　DVI を可読テキストに変換
gftodvi　Generic fonts（GF）の見本（DVI）作成
gftopk　GF を packed fonts（PK）に変換
gftype　GF を可読テキストに変換
mf　フォントファミリの作成
mft　METAFONT ソースを TEX ファイルに変換
mpost　図形描画
patgen　ハイフンパターンを作成
pktogf　PK を GF に変換
pktype　PK を可読テキストに変換
pltotf　プレーンテキストのプロパティリストを TFM に変換
pooltype　WEB プールファイルを表示
tangle　WEB を Pascal に変換
tex　組版システム
tftopl　TFM をプレーンテキストのプロパティリストに変換
vftovp　仮想フォントを仮想プロパティリストに変換

vptovf 仮想プロパティリストを仮想フォントに変換
weave WEB を TEX ファイルに変換

各プログラムの正確な機能やシンタックスについては、それぞれのマニュアルや Web2C 自体のドキュメントに記載されています。しかしながら、すべてのプログラムに共通するいくつかの基本事項を押さえておくと Web2C をインストールする際に役立ちます。

Web2C に含まれるすべてのプログラムは、以下の標準的な GNU オプションをサポートしています：

`--help` 基本的な使用法を表示
`--version` バージョン情報を表示して終了

さらに、ほとんどが以下のオプションもサポートしています：

`--verbose` より詳細な実行メッセージを表示

Web2C プログラムはファイル探索に Kpathsea ライブラリ（https://tug.org/kpathsea）を使用します。このライブラリは環境変数と設定ファイルの両方を参照して膨大な量の TEX 関連ファイルの検索を最適化します。Web2C は多くの TEXMF ツリーを同時に参照でき、このことが TDS を維持したままローカルシステム全体またはユーザごとの拡張を異なるツリーに分離するのに役立ちます。ファイル検索を高速化するため、各ツリーのルートには配下にあるすべてのファイルの名前と相対位置を列挙した `ls-R` ファイルが置かれています。

7.1 Kpathsea パス検索

まずはじめに、Kpathsea ライブラリによるパス検索の一般的なメカニズムを紹介します。

パス要素（基本的にはディレクトリ名）をコロンまたはセミコロン区切りで並べたリストを**検索パス**と呼びます。検索パスはさまざまな場所で設定可能です。例えば、‘`my-file`’ を検索パス ‘`.:/dir`’ から見つける場合、Kpathsea は各パス要素を順番に確認します。この場合、まず `./my-file` を確認し、次に `/dir/my-file` を確認して最初にみつけたもの（状況によっては、見つけたものすべて）を返します。

あらゆる OS に適切に対応するため、非 Unix 系システムにおいては Kpathsea はコロン（‘`:`’）とスラッシュ（‘`/`’）以外の記号をファイル名の区切り文字として扱うことができ

texlive-ja (2024-03-10 11:49)

ます。

特定のパス要素 p を確認する際、Kpathsea はまず利用可能なファイル名データベース（7.2 節）が p について用意されているかを調べます。すなわち、p のプレフィックスにあたるディレクトリにデータベースがあるかを調べ、もし存在していた場合はその内容に基づいて検索マッチングを行います。

最もシンプルでよくあるパス要素はディレクトリ名ですが、Kpathsea はそれ以外の機能として階層的なビルトイン値、環境変数名、設定ファイルの値、ユーザのホームディレクトリ、そしてサブディレクトリの再帰的な検索などをサポートしています。すなわち、Kpathsea はパス要素を種々の指定に従って通常のディレクトリ名に**展開**します。これ以降、この展開について実際に処理が行われる順番で説明していきます。

ところで、ファイル名が絶対パスもしくは明示的な相対パス（すなわち ‘/’、‘./’、‘../’ のいずれかから始まるもの）で検索された場合は Kpathsea は、単純にそのパス位置にファイルがあるか否かを確認します。

7.1.1 パスの設定元

検索パスは設定ファイルや環境変数など、さまざまな場所で設定することができます。ここにすべての設定元を Kpathsea が使用する順番に列挙しておきます：

1. ユーザが設定している `TEXINPUTS` などの環境変数。なお ‘⟨変数名⟩.⟨プログラム名⟩’ という形の環境変数は、環境変数 ‘⟨変数名⟩’ を上書きします。例えば ‘`latex`’ が実行中のプログラム名である場合、`TEXINPUTS.latex` は `TEXINPUTS` を上書きします。
2. プログラムごとの設定ファイル。例えば dvips の設定ファイル `config.ps` 内にある ‘`S /a:/b`’ のような記述が反映されます。
3. Kpathsea の設定ファイル `texmf.cnf`。‘`TEXINPUTS=/c:/d`’ のような設定が並んでいます。
4. コンパイル時のデフォルト設定。

それぞれの変数の値が何に設定されているかは、デバッグオプションを使用すると確認することができます（p. 63 の「デバッグアクション」項を参照）。

7.1.2 設定ファイル

Kpathsea は `texmf.cnf` という名前の実行時設定ファイルを読み込み、検索パスやその他の設定を行います。この設定ファイルの検索には検索パス `TEXMFCNF` が使用されます

が、この変数の内容を変更したりシステムディレクトリを上書きするように環境変数の値を設定することは推奨されません。

代わりに、通常のインストールを行うと .../2024/texmf.cnf が生成されます。どうしてもデフォルト設定を変更する必要が生じた場合は（通常は必要ありません）、このファイルに変更を加えてください。メインの設定ファイルは .../2024/texmf-dist/web2c/texmf.cnf です。このファイルを変更してもディストリビューション側でアップデートがあると変更内容が失われてしまうため、ユーザはこのファイルを編集すべきではありません。

余談ですが、もし単に個人用のディレクトリを特定の検索パスに入れたいという場合は環境変数を設定するのが合理的です：

```
TEXINPUTS=.:/my/macro/dir:
```

設定をメンテナンスしやすく、また何年にも亘って使えるようにするためには、すべての要素を明示的に記入するのではなく、末尾に ‘:’（Windows では ‘;’）を置いてシステムのパスが挿入されるようにしましょう（7.1.4 節を参照）。また、他にも TEXMFHOME ツリーを利用する方法もあります（3.2.3 節を参照）。

Kpathsea 検索パスに含まれる**すべて**の texmf.cnf を読み込み、その内容を反映します。この際、検索パスのより前方にあるものの設定はより後方にあるものよりも優先されます。例えば、検索パス .:$TEXMF の下で、./texmf.cnf は $TEXMF/texmf.cnf の内容を上書きします。

- % と # はコメント開始文字で、行頭に置くとその行の末尾までコメントとして扱われます。% の前に空白文字があっても同様です。
- 空行は無視されます。
- 行末の \ は行継続文字としてはたらきます。すなわち、次の行の内容も同じ行として扱われます。行頭の連続する空白文字は無視されます。
- ほかの行は次の形式で記述されます：

  ⟨変数名⟩[.⟨プログラム名⟩] [=] ⟨値⟩

  なお ‘=’ とその前後の空白は省略可能です（ただし ⟨値⟩ が ‘.’ で始まる場合は、ピリオドがプログラム修飾子と解釈されないよう ‘=’ を書いた方が無難です）。
- ⟨変数名⟩ には空白文字、‘=’、‘.’ 以外の任意の文字を含めることができますが ‘A-Za-z_’ の範囲に留めておくのが無難です。

- もし変数名に '.⟨プログラム名⟩' が付いている場合は、その変数定義は ⟨プログラム名⟩ または ⟨プログラム名⟩.exe という名前のプログラムの実行中のみで適用されます。この機能を用いると、例えば TEX 処理系の種類ごとに異なる検索パスを適用することができます。
- 文字列とみなされるため、value にはあらゆる文字を含めることができます。しかし、実際には texmf.cnf に記述されるほとんどの変数はパス展開に関係し、そのパス展開においてはさまざまな特殊文字が使用される（7.1.7 節を参照）ため、ディレクトリ名としてはブレースやコンマを使用することはできません。
  また変数中の ';' は Unix 系の環境においては ':' に読み替えられます。これは単一の texmf.cnf ファイルで Unix 系と Windows の双方をサポートするための措置です。この読み替えは検索パスに限らずすべての変数について適用されますが、幸い ';' は検索パス以外の変数には必要のない文字です。
  なお $var.prog という表記は右辺ではサポートされません。代替手段として、そのようなことをしたい場合には新しい変数を追加してください。
- すべての定義は展開される前に読み込まれるので、定義前の変数も右辺に含めることが可能です。

上記の仕様のほとんどを活用した設定ファイルの例を示しておきます：

```
TEXMF              = {$TEXMFLOCAL,!!$TEXMFMAIN}
TEXINPUTS.latex    = .;$TEXMF/tex/{latex,generic;}//
TEXINPUTS.fontinst = .;$TEXMF/tex//;$TEXMF/fonts/afm//
% e-TeX related files
TEXINPUTS.elatex   = .;$TEXMF/{etex,tex}/{latex,generic;}//
TEXINPUTS.etex     = .;$TEXMF/{etex,tex}/{eplain,plain,generic;}//
```

7.1.3 パス展開

Kpathsea は Unix シェルと同様にいくつかの特殊文字や検索パスの構造を認識します。一般の例として、パス ~$USER/{foo,bar}//baz は $USER のホームディレクトリ直下のディレクトリ foo と bar 以下にある、baz という名前のファイルまたはディレクトリを含むすべてのサブディレクトリに展開されます。これらのパス展開の詳細については以下のセクションを参照してください。

7.1.4 デフォルト展開

最も優先度の高い検索パス（7.1.1 節を参照）が先頭、末尾または途中 2 連続する形で**追加コロン**を含んでいる場合、Kpathsea はその位置に（もし定義されていれば）次に優先度の高いパスを挿入します。もし挿入される次点の検索パスにも追加コロンが含まれている場合、さらに次に優先度の高いパスが挿入されていきます。例えば

```
> setenv TEXINPUTS /home/karl:
```

のように環境変数が設定されていたとして、`texmf.cnf` で設定されている `TEXINPUTS` の値が

```
.:$TEXMF//tex
```

のとき、最終的に検索されるパスは次のようになります：

```
/home/karl:.:$TEXMF//tex
```

デフォルト値を複数箇所に挿入しても意味がないので、Kpathsea は追加コロンを 1 つだけデフォルト値に置換し、それ以外はそのままにします。まず先頭にコロンがないか確認され、続いて末尾コロン、2 連続のコロンが順にチェックされます。

7.1.5 ブレース展開

ブレース展開は便利な機能で、例えば `v{a,b}w` は `vaw:vbw` に展開されます。ネストすることも可能です。この機能は `$TEXMF` にブレースリストを与えることにより複数の TEX ヒエラルキーを共存させるのにも利用されています。具体的には、デフォルトの `texmf.cnf` では次のような設定が行われています（この例のために単純化しています）：

```
TEXMF = {$TEXMFVAR,$TEXMFHOME,!!$TEXMFLOCAL,!!$TEXMFDIST}
```

この値は、例えば TEX の入力パスを定義するのに活用できるでしょう：

```
TEXINPUTS = .;$TEXMF/tex//
```

この設定の下では、まずカレントディレクトリがチェックされ、続いて `$TEXMFVAR/tex`, `$TEXMFHOME/tex`, `$TEXMFLOCAL/tex`, `$TEXMFDIST/tex` ツリーが（最後 2 つについては `ls-R` データベースを利用しつつ）検索されることになります。

7.1.6 サブディレクトリ展開

パス要素内で、2 つ以上の連続するスラッシュがディレクトリ名 d の後ろに置かれている場合、d 以下のすべてのサブディレクトリに展開されます。つまり、まず d 直下のサブディレクトリが並び、続いてさらにそれらのサブディレクトリのサブディレクトリが並び、以降も再帰的に続きます。各階層でサブディレクトリがどういう順番で検索されるかは決まっていません。

'//' に続いてファイル名の一部を置いた場合、それとマッチするファイルを含むサブディレクトリのみが展開結果に含められます。例えば '/a//b' は /a/1/b, /a/2/b, /a/1/1/b などに展開されますが、/a/b/c や /a/1 には展開されません。

'//' を 1 つのパスに複数含めることは可能ですが、パスの先頭に置かれた '//' は無視されます。

7.1.7 texmf.cnf で使用できる特殊文字の概要

以下のリストに Kpathsea 設定ファイルで利用できる特殊文字と構造について要約しておきます：

:	パスの先頭や末尾に書かれた場合および途中で 2 つ続けて書かれた場合、その部分はデフォルトパスの展開形で置き換えらる
;	非 Unix 系システムでの区切り文字（挙動は : と同様）
$	変数展開
~	ユーザのホームディレクトリを表す
{...}	ブレース展開
,	ブレース内における要素の区切り文字
//	サブディレクトリ展開。先頭を除き、パス中どこでも利用可能
% と #	コメントの開始
\	行末に置くと、行継続文字としてはたらき複数行入力を可能にする
!!	ファイル名データベースの使用を強制（ディスク検索を禁止）

これらの文字が、正確にどのタイミングで特殊なものとして扱われるのか、あるいはその文字そのものとして扱われるのかは文脈に依存します。設定解釈の各段階の処理（パース、展開、検索など）と不可分に結びついているため、残念ながら簡潔に説明することはできません。また、汎用のエスケープ手段も存在していません。特に '\' は texmf.cnf 内においては「エスケープ文字」ではないので注意してください。

インストールを行う際にディレクトリ名を選ぶ際には、すべての特殊文字を含めないようにするのが無難です。

7.2 ファイル名データベース

Kpathsea は検索のためのディスクアクセスをある程度軽減する仕組みを持っています。それでも、標準的な TEX Live やその他十分な大きさの TEX インストール構成において、指定されたファイルについて、それが存在する可能性のあるすべてのディレクトリを検索するとかなり長い時間がかかってしまいます。そのため Kpathsea は `ls-R` という名前のプレインテキスト「データベース」にファイルとディレクトリの情報を予め書き込んでおき、過剰なディスクアクセスを回避しています。

また `aliases` というもう 1 つのデータベースファイルを利用すると、`ls-R` に含まれているファイル名に別名（エイリアス）を付けることができます。

7.2.1 ファイル名データベース

上で説明したように、メインのファイル名データベースの名前は `ls-R` です。ファイル名データベースは、検索対象としたい各 TEX ヒエラルキーの最上位階層に配置することができます（デフォルトでは `$TEXMF`）。Kpathsea は `TEXMFDBS` パスに含まれている `ls-R` を読み込みます。

`ls-R` の生成・更新には、ディストリビューションに含まれる `mktexlsr` コマンドの利用をおすすめします。このコマンドは種々の `mktex...` スクリプトから呼び出されます。本質的には、このスクリプトは単に

```
cd /your/texmf/root && \ls -1LAR ./ >ls-R
```

を実行するだけです。なお、このスクリプトはシステムの `ls` コマンドが正しい形式で出力を行うことを前提にしています（GNU `ls` なら問題ありません）。データベースが常に最新の状態であることを保証するには、`cron` を利用して定期的にデータベースを更新するのが簡単でしょう。そうすることによって、例えば LATEX パッケージをインストールしたりアップデートしたりして、新しいファイルが配置されたとしても自動的にデータベースが更新されることになります。

データベースにファイルが見つからない場合、デフォルトでは Kpathsea は処理を続行してディスク検索を行います。ただし '`!!`' から開始するパス要素については、データベースのみが検索され、ディスクが検索されることはありません。

7.2.2 kpsewhich: パス検索用コマンドラインツール

kpsewhich コマンドはパス検索を行うための独立したツールです。このコマンドは find コマンドのように利用して TEX ヒエラルキーの中にあるファイルを探索するのに便利です（このプログラムはディストリビューションに含まれる mktex... スクリプトでも多用されています)。

> kpsewhich ⟨オプション⟩... ⟨ファイル名⟩...

上記の ⟨オプション⟩ は '-' または '--' で開始して、曖昧でない限りどのような省略の仕方をしても指定可能です。

Kpathsea はオプションでないすべての引数をファイル名として受け取って、それぞれについて見つかった最初のファイルを返します。特定の名前のファイルをすべて列挙するオプションはありません（そのような場合は Unix の find コマンドを使用してください)。

よく使われるオプションを一覧にしておきます：

--dpi=⟨数値⟩ 解像度を ⟨数値⟩ に設定します。このオプションは generic fonts（GF）と packed fonts（PK）を検索するときにのみ効果を発揮します。dvips に合わせて -D という省略形も使用できます。デフォルト値は 600 です。

--format=⟨フォーマット名⟩ 検索対象のフォーマットを指定します。デフォルトではフォーマットはファイル名から推測されます。METAPOST のサポートファイルや dvips の設定ファイルなど、対応する（曖昧でない）拡張子がないフォーマットについては、このオプションを用いて明示的に指定する必要があります。その際 ⟨フォーマット名⟩ には tex や enc files などを指定します。具体的なフォーマット名のリストについては kpsewhich --help-formats を参照してください。

--mode=⟨モード名⟩ モード名を設定します。このオプションも GF および PK の検索用です。デフォルト値はなく、指定しない限りすべてのモードが検索対象になります。

--must-exist ディスク検索を含むあらゆる手段を使用して、可能な限りファイル検索を行います。デフォルトでは、効率重視のため ls-R データベースのみが検索されます。

--path=⟨パス⟩ ファイル名からパスを推測する代わりに、指定された ⟨パス⟩（通常通りコロン区切り）を利用して検索を行います。'//' を含むすべての特殊文字とその展

開がサポートされています。--path と --format は相容れません。

--progname=⟨プログラム名⟩ プログラム名を設定します。これは .⟨プログラム名⟩ 接尾辞付きの変数を介して検索パスに影響を与えます。デフォルトでは kpsewhich です。

--show-path=⟨名前⟩ 指定した ⟨名前⟩ と紐づくファイルタイプの検索に用いられるパスを表示します。⟨名前⟩ には拡張子（.pk, .vf, etc.）またはファイル名を指定することができ、--format オプションも併用できます。

--debug=⟨数値⟩ デバッグレベルを ⟨数値⟩ に設定します。

7.2.3 使用例

このセクションでは Kpathsea の実行例をいくつか紹介していきます。まずは単純な検索です：

```
> kpsewhich article.cls
  /usr/local/texmf-dist/tex/latex/base/article.cls
```

この例では article.cls を検索しています。拡張子 '.cls' には曖昧性がないので、わざわざファイルタイプ tex（TEX のソースファイル）を指定する必要はありません。上の例では、目的のファイルは texmf-dist TEX Live ディレクトリ内の tex/latex/base サブディレクトリの中に見つかりました。同様にして、以下に列挙する例はいずれも曖昧性のない拡張子のおかげで特に問題なくファイルを見つけることができます：

```
> kpsewhich array.sty
  /usr/local/texmf-dist/tex/latex/tools/array.sty
> kpsewhich latin1.def
  /usr/local/texmf-dist/tex/latex/base/latin1.def
> kpsewhich size10.clo
  /usr/local/texmf-dist/tex/latex/base/size10.clo
> kpsewhich small2e.tex
  /usr/local/texmf-dist/tex/latex/base/small2e.tex
> kpsewhich tugboat.bib
  /usr/local/texmf-dist/bibtex/bib/beebe/tugboat.bib
```

ところで、最後の例として挙がっているのは *TUGboat* の記事用の BIBTEX 文献データ

ベースです。

```
> kpsewhich cmr10.pk
```

拡張子 '.pk' をもつ、フォントのビットマップグリフファイルは dvips や xdvi などのディスプレイプログラムによって使用されます。Computer Modern の '.pk' ファイルは通常 TEX Live に含まれていないので（デフォルトでは Type 1 変種が使用されるため)、上の例では何も出力されません。

```
> kpsewhich wsuipa10.pk
  /usr/local/texmf-var/fonts/pk/ljfour/public/wsuipa/wsuipa10.600pk
```

これらのフォント（ワシントン大学で開発された音声記号）については、まず '.pk' ファイルを生成する必要があり、TEX Live における METAFONT モードはデフォルトでは解像度 600 dpi（dots per inch）設定の ljfour なので、上のような結果になります。

```
> kpsewhich -dpi=300 wsuipa10.pk
```

続いてのケースでは解像度 300 dpi を指定（-dpi=300）していますが、そのようなフォントファイルは存在しないという結果が得られました。dvips や xdvi などのプログラムは実行を中断し、mktexpk スクリプトを用いて必要な '.pk' ファイルを生成することになるでしょう。

次に dvips のヘッダおよび設定ファイルについて考えましょう。2004 年以来、汎用の設定ファイル config.ps と PostScript フォントのマップファイル psfonts.map は独自の検索パスと texmf ツリー内での配置場所を持っているので、それらを取り扱う前に TEX サポートのための汎用プロローグ tex.pro の例を示しました。拡張子 '.ps' には曖昧性があるので、config.ps については該当するファイルタイプ（dvips config）を明示的に指定する必要があります。

```
> kpsewhich tex.pro
  /usr/local/texmf/dvips/base/tex.pro
> kpsewhich --format="dvips config" config.ps
  /usr/local/texmf/dvips/config/config.ps
> kpsewhich psfonts.map
  /usr/local/texmf/fonts/map/dvips/updmap/psfonts.map
```

今度は URW Times の PostScript サポートファイルについて考えます。標準的なフォ

ント命名規則に基づけば、これらのプレフィックスは 'utm' です。最初に検索するのはマップファイルの名前を格納している設定ファイルです：

```
> kpsewhich --format="dvips config" config.utm
  /usr/local/texmf-dist/dvips/psnfss/config.utm
```

このファイルには

```
  p +utm.map
```

という記述があり、utm.map を参照しているので、続いてこのファイルの位置を確認してみましょう：

```
> kpsewhich utm.map
  /usr/local/texmf-dist/fonts/map/dvips/times/utm.map
```

このマップファイルは URW コレクションに含まれる Type 1 PostScript フォントのファイル名を定義しています。ファイルには下記のような記述が並んでいます（以下は抜粋です）：

```
utmb8r  NimbusRomNo9L-Medi    ... <utmb8a.pfb
utmbi8r NimbusRomNo9L-MediItal... <utmbi8a.pfb
utmr8r  NimbusRomNo9L-Regu    ... <utmr8a.pfb
utmri8r NimbusRomNo9L-ReguItal... <utmri8a.pfb
utmbo8r NimbusRomNo9L-Medi    ... <utmb8a.pfb
utmro8r NimbusRomNo9L-Regu    ... <utmr8a.pfb
```

例えば、Times Roman を例にとって、texmf ディレクトリ内における utmr8a.pfb ファイルの位置を検索してみます：

```
> kpsewhich utmr8a.pfb
  /usr/local/texmf-dist/fonts/type1/urw/times/utmr8a.pfb
```

ここまでの例が、いかに簡単に指定したファイルを見つけることができるかを示しているはずです。ここで示したテクニックは、何らかの理由で間違ったバージョンのファイルが使用されてしまっていると疑われるときに特に役立ちます。kpsewhich コマンドを用いれば、どのファイルが一番最初に見つかっているのかを確認することができます。

7.2.4 デバッグアクション

ときとして、あるプログラムがファイル参照をどのように解決しているかについて調査する必要が生じる場合があります。そのような場合のために、Kpathsea はさまざまなレベルのデバッグオプションを提供しています：

1 stat コマンドによるディスク検索の状況。きちんと更新された ls-R データベースを使用して検索が行われている場合、このデバッグオプションはほとんど何も出力しないはずです。
2 ハッシュテーブルの参照（ls-R、マップファイル、設定ファイルなど）。
4 ファイル I/O 操作。
8 各ファイルタイプに適用されるパスの情報。これは特定のファイルに対する検索パスがどこで定義されているのかを調べるのに役立ちます。
16 各パス要素に属するディレクトリの一覧（ディスク検索にのみ関係）。
32 ファイル検索。
64 変数の値。

デバッグレベルを -1 に設定すると上のすべてが有効になります。多くの場合、実用上はこれが一番便利でしょう。

同様に dvips プログラムでも、デバッグオプションをいくつか組み合わせることでどこのファイルが読み込まれているか詳細を確認することができます。またファイルが見つからない場合に、デバッグ出力はプログラムがどこに対してファイル検索を行ったかを示してくれるので、どこに問題があるのかを把握することができます。

一般的に言って、Kpathsea ライブラリは多くのプログラムから内部的に呼び出されるものなので、デバッグオプションは環境変数 KPATHSEA_DEBUG を上記リストの値（もしくはそれらの組み合わせ）に設定することによって指定するのがよいでしょう。[*13]

例として、以下のような内容の小さな LaTeX ソースファイル hello-world.tex を考えてみます。

```
\documentclass{article}
\begin{document}
```

[*13] Windows ユーザ向けの情報：Windows ではすべてのデバッグメッセージを 1 つのファイルにリダイレクトするのは困難です。デバッグの際は、一時的に SET KPATHSEA_DEBUG_OUTPUT=err.log などと設定するのがよいでしょう。

```
debug:start search(file=texmf.cnf, must_exist=1, find_all=1,
  path=.:/usr/local/bin/texlive:/usr/local/bin:
       /usr/local/bin/texmf/web2c:/usr/local:
       /usr/local/texmf/web2c:/.:/./teTeX/TeX/texmf/web2c:).
kdebug:start search(file=ls-R, must_exist=1, find_all=1,
  path=~/tex:/usr/local/texmf).
kdebug:search(ls-R) =>/usr/local/texmf/ls-R
kdebug:start search(file=aliases, must_exist=1, find_all=1,
  path=~/tex:/usr/local/texmf).
kdebug:search(aliases) => /usr/local/texmf/aliases
kdebug:start search(file=config.ps, must_exist=0, find_all=0,
  path=.:~/tex:!!/usr/local/texmf/dvips//).
kdebug:search(config.ps) => /usr/local/texmf/dvips/config/config.ps
kdebug:start search(file=/root/.dvipsrc, must_exist=0, find_all=0,
  path=.:~/tex:!!/usr/local/texmf/dvips//).
search(file=/home/goossens/.dvipsrc, must_exist=1, find_all=0,
  path=.:~/tex/dvips//:!!/usr/local/texmf/dvips//).
kdebug:search($HOME/.dvipsrc) =>
kdebug:start search(file=config.cms, must_exist=0, find_all=0,
  path=.:~/tex/dvips//:!!/usr/local/texmf/dvips//).
kdebug:search(config.cms)
=>/usr/local/texmf/dvips/cms/config.cms
```

図 12　設定ファイルを見つける

```
Hello World!
\end{document}
```

この小さなファイルは cmr10 フォントのみを使用するのですが、dvips がどのように PostScript ファイルを用意するのか確認してみましょう（ここでは Computer Modern の Type 1 バージョンを使用したいので -Pcms を指定します）。

```
> dvips -d4100 hello-world -Pcms -o
```

上記の例では dvips のデバッグクラス 4（フォントパス）と Kpathsea のパス要素展開を組み合わせました（dvips のリファレンスマニュアルを参照）。この出力（かなりいじっていますが）を図 12 に示します。dvips は起動後、まず作業ファイルを探します。はじめに他のファイルの検索に用いられる検索パスを定義する texmf.cnf が見つかり、続いてファイル検索を最適化するファイル名データベース ls-R とエイリアスを定義してい

る aliases が見つかります。dvips はさらに汎用の設定ファイル config.ps やカスタマイズ設定を記述する .dvipsrc を探し出します（上の例では、後者は “not found” という結果になっています)。最後に dvips は Computer Modern の PostScript フォント用の設定ファイル config.cms を探します（これは -Pcms の効果です)。このファイルには TEX 間の関係を定義するマップファイルのリストが記述されています。

```
> more /usr/local/texmf/dvips/cms/config.cms
   p +ams.map
   p +cms.map
   p +cmbkm.map
   p +amsbkm.map
```

dvips はこのようにしてすべてのファイルを見つけ、さらに常に使用されることになっている汎用マップファイル psfonts.map も読み込みます（このファイルではすべての PostScript フォントに共通する定義が行われています。PostScript マップファイルの取り扱いの詳細については 7.2.3 節の最後の部分を参照してください)。

ここにきて、dvips はやっとプログラム自身の情報をユーザに提示します：

```
This is dvips(k) 5.92b Copyright 2002 Radical Eye Software
```

その後プロローグファイルが検索されます：

```
kdebug:start search(file=texc.pro, must_exist=0, find_all=0,
  path=.:~/tex/dvips//:!!/usr/local/texmf/dvips//:
       ~/tex/fonts/type1//:!!/usr/local/texmf/fonts/type1//).
kdebug:search(texc.pro) => /usr/local/texmf/dvips/base/texc.pro
```

以上で必要なファイルが揃ったので、dvips は日付と時刻を出力し、そして hello-world.ps というファイルが生成されることを予告します。併せて、そのファイル生成の際にはフォントファイル cmr10 が必要であることと、そのフォントが “resident” （ビットマップ不要）として扱われる旨が表示されます。

```
TeX output 1998.02.26:1204’ -> hello-world.ps
Defining font () cmr10 at 10.0pt
Font cmr10 <CMR10> is resident.
```

これでやっと cmr10.tfm の検索が行われ、それが見つかれば、もういくつかのプロローグファイルが参照され（出力には出ていません)、最終的に Type 1 のフォント実体

cmr10.pfb が出力ファイルに含まれることになります（最終行）。

```
kdebug:start search(file=cmr10.tfm, must_exist=1, find_all=0,
  path=.:~/tex/fonts/tfm//:!!/usr/local/texmf/fonts/tfm//:
       /var/tex/fonts/tfm//).
kdebug:search(cmr10.tfm) => /usr/local/texmf/fonts/tfm/public/cm/cmr10.tfm
kdebug:start search(file=texps.pro, must_exist=0, find_all=0,
   ...
<texps.pro>
kdebug:start search(file=cmr10.pfb, must_exist=0, find_all=0,
  path=.:~/tex/dvips//:!!/usr/local/texmf/dvips//:
       ~/tex/fonts/type1//:!!/usr/local/texmf/fonts/type1//).
kdebug:search(cmr10.pfb) => /usr/local/texmf/fonts/type1/public/cm/cmr10.pfb
<cmr10.pfb>[1]
```

7.3 ランタイムオプション

その他の Web2C の便利な機能として、数々のメモリパラメタ（例えば配列サイズ）を Kpathsea が読み込む設定ファイル texmf.cnf でコントロールできることが挙げられます。メモリ設定は TEX Live ディストリビューションに含まれる texmf.cnf の Part 3 に記述されています。より重要なことは：

main_memory TEX, METAFONT, METAPOST に使用を許可する合計のメモリ量（ワード数）。これらの設定を反映するには、値を適用した上でそれぞれのフォーマットを生成する必要があります。例えば、もし「巨大な」バージョンの TEX を生成したとして、それを利用するにはフォーマットファイル hugetex.fmt を呼び出す必要があります。

extra_mem_bot「大きな」TEX のデータ構造（ボックス、グルー、ブレイクポイントなど）を配置するための追加領域。特に PICTEX を使用する際に便利です。

font_mem_size フォント情報を格納するために使用するメモリ量（ワード数）。これは読み込まれる TFM ファイルの合計サイズと概ね一致します。

hash_extra 制御綴を格納するハッシュテーブルのための追加領域。デフォルト値は 600000 です。

こうしたパラメタの存在は動的な配列・メモリ確保の完全な代替となるものではありません。しかし、そうした変更を現在の TEX ソースに対して適用するのは容易でないので、

これらの実行時パラメタの存在はある程度の柔軟性をもたせる実用上の妥協策です。

7.4 $TEXMFDOTDIR

ここまでさまざまな箇所で、(カレントディレクトリを最初に検索するようにするため) ドット '.' から始まる検索パスを紹介してきました：

```
TEXINPUTS=.;$TEXMF/tex//
```

実は、これは単純化した表記です。TEX Live が提供している実際の texmf.cnf では単に '.' と書く代わりに $TEXMFDOTDIR を使用しています：

```
TEXINPUTS=$TEXMFDOTDIR;$TEXMF/tex//
```

(実際のファイルでは、2 つ目のパス要素も $TEXMF/tex// よりは少し複雑です。とはいえ、それは些末なことなので、ここでは $TEXMFDOTDIR の話に注目してください。)

パス定義において、単に '.' と書く代わりに $TEXMFDOTDIR を使用する理由は、単純にその値を上書きできるメリットがあるからです。例えば、複雑なドキュメントは数多くのソースファイルが、多くのサブディレクトリにまたがって配置されている場合があります。そのような場合、TEXMFDOTDIR の値を .// に（例えば、そのドキュメントをビルドする環境で）設定するだけで、サブディレクトリのすべてが検索されるようになります。(警告：.// をデフォルトの設定にはしないでください。そうした設定は極めて好ましくなく、任意のドキュメントについてすべてのサブディレクトリを検索するようにすることには潜在的なセキュリティ上の問題があります。)

また別のケースでは、カレントディレクトリを一切検索したくないという場合も考えられます。それは例えば、すべてのファイルを明示的なパスで記述しているような状況で起こり得ます。そのような場合、TEXMFDOTDIR の値は /nonesuch や任意の存在しないディレクトリに設定しておくとよいでしょう。

$TEXMFDOTDIR のデフォルト値は、TEX Live の texmf.cnf に定義されているように単に '.' です。

8 謝辞

TEX Live は事実上すべての TEX ユーザ会の貢献に支えられています。現在の TEX Live 監修者は Karl Berry です。過去から現在に至るまでの、その他の主要な貢献者は以下の通りです（敬称略）：

- 英語圏、ドイツ、オランダ、ポーランドの TEX ユーザ会（それぞれ TUG, DANTE e.V., NTG, GUST）。技術的・経営的なインフラの提供してくれました。ぜひお近くの TEX ユーザ会に加入してください！（https://tug.org/usergroups.html を参照）
- CTAN チーム（urlhttps://ctan.org）。TEX Live イメージを配布し、TEX Live が依存する各種のパッケージアップデートを行うための共通基盤の提供をしてくれています。
- Nelson Beebe。TEX Live 開発陣に数々のプラットフォームを提供し、彼自身も多くの包括的なテストを行ってくれました。また彼の比類なき書誌学的な貢献にも敬意を表します。
- John Bowman。彼の高度なグラフィックツール Asymptote を TEX Live に対応させるため数多くの変更を加えてくれました。
- Peter Breitenlohner と ε-TEX チーム。将来の TEX のために安定した基盤を構築してくれました。特に Peter 氏は、長年に渡って GNU autotools 関連のヘルパーとして活躍し、ソースコードを最新の状態に保ってくれました。Peter。2015 年 10 月に亡くなりましたが、私たちは彼のことを忘れません。
- Jin-Hwan Cho とすべての DVIPDFMx チームメンバー。素晴らしいドライバを開発し、設定周りの諸問題に対する責任ある対応をしてくれました。
- Thomas Esser。彼の素晴らしい teTEX パッケージなしには TEX Live は存在し得ませんでした。
- Michel Goossens。この文書のオリジナル版の共著者です。
- Eitan Gurari。彼の TEX4ht は TEX Live 公式ドキュメントの HTML 版生成に利用されています。彼は急な依頼をしても、毎年快く TEX4ht の改善に取り組んでくれました。Eitan は 2009 年 6 月に若くして亡くなりましたが、私たちは彼のことを忘れません。
- Hans Hagen。多くのテストを引き受け、彼の ConTEXt パッケージ（http://pragma-ade.com）を TEX Live フレームワークに対応させるとともに、継続的に TEX 開発に協力してくれました。
- Hàn Thế Thành, Martin Schröder と pdfTEX チーム（https://tug.org/applications/pdftex/）。彼らは TEX を継続的に拡張してくれています。
- Hartmut Henkel。pdfTEX, LuaTEX をはじめ多くのプロジェクトに顕著な貢献をしてくれました。
- 平田俊作。DVIPDFMx について多くの新規および継続的な貢献をしてくれま

した。

- Taco Hoekwater。METAPOST と (Lua)TEX（https://luatex.org）自体に対して大きな変更を施したほか、ConTEXt の TEX Live 対応や Kpathsea のマルチスレッド機能、その他数々の貢献をしてくれました。
- Khaled Hosny。XƎTEX, DVIPDFMx に加え、アラビア語用のものを含め多くのフォントに対して多くの貢献をしてくれました。
- Paweł Jackowski。Windows 用インストーラ `tlpm`、Tomasz Łuczak 氏は `tlpmgui` を開発してくれました。これらのプログラムは過去の TEX Live で使用されていました。
- 角藤亮。日本語 TEX 用の W32TEX, W64TEX ディストリビューション（http://w32tex.org）のために Windows バイナリを提供し、さらに開発に関わる数々の貢献をしてくれました。
- Jonathan Kew。優れた TEX エンジンである XƎTEX を開発し、時間と労力を割いてその処理系を TEX Live に導入してくれました。彼はさらに初期バージョンの MacTEX を開発し、我々のおすすめするフロントエンドである TEXworks も開発しました。
- 北川弘典。(ε-)pTEX のメンテナンスと関連する多くのサポートを担ってくれました。
- Dick Koch。TEX Live チームと友好的かつ緊密に連携しながら MacTEX（https://tug.org/mactex）のメンテナンスを行ってくれています。
- Reinhard Kotucha。TEX Live 2008 のインフラとインストーラに対する主要な貢献者で、さらに Windows 研究や `getnonfreefonts` スクリプトの開発などに多大な努力をしてくれています。
- Siep Kroonenberg。彼も TEX Live 2008 のインフラとインストーラの（特に Windows について）主要貢献者で、その新機能について解説するためマニュアルを大幅に改定してくれました。
- Clerk Ma。エンジンのバグ修正と拡張を行ってくれました。
- Mojca Miklavec。ConTEXt 開発の重要な協力者で、多くのバイナリビルドを行い、その他にも多くの仕事を行ってくれています。
- Heiko Oberdiek。epstopdf をはじめ数々のパッケージを開発し、巨大だった pstgeo データファイルを圧縮して TEX Live に含められるようにしてくれました。そして何より、彼の hyperref パッケージについての偉業に敬意を表します。
- Phelype Oleinik。2020 年にグループ区切りでの `\input` 対応を多くのエンジンに

適用したほか、多くの貢献をしてくれました。

- Petr Olšak。チェコ・スロバキア関連のプロダクトを統括し、注意深くチェックしてくれました。
- 大島利雄。Windows 向け DVI ビューア `dviout` を開発してくれました。
- Manuel Pégourié-Gonnard。パッケージアップデートとドキュメントの改善に協力してくれたほか、`texdoc` コマンドを開発しました。
- Fabrice Popineau。はじめて TeX Live の Windows サポートを行い、またフランス語ドキュメントに貢献してくれました。
- Norbert Preining。現在の TeX Live インフラとインストーラの主要開発者で、Frank Küster と共に TeX Live の Debian パッケージを用意してくれています。彼は現在も広範に TeX Live に関わるさまざまな仕事を引き受けてくれています。
- Sebastian Rahtz。TeX Live の創始者で、長年そのメンテナンスを行っていました。彼は 2016 年 3 月に亡くなりましたが、私たちは彼のことを忘れません。
- Luigi Scarso。METAPOST, LuaTeX その他のプログラムを継続的に開発してくれています。
- Andreas Scherer。TeX Live で用いている CWEB 実装である `cwebbin` の開発およびオリジナル CWEB の継続的なメンテナンスを行ってくれました。
- 田中琢爾。(ε-)(u)pTeX のメンテナンスと関連する多くのサポートを担ってくれました。
- Tomasz Trzeciak。Windows について広範な協力をしてくれました。
- Vladimir Volovich。さまざまなプログラムの TeX Live への組み込みとメンテナンスのために多くの手伝いをしてくれました。特に `xindy` の組み込みが実現したのは彼のおかげです。
- Staszek Wawrykiewicz。TeX Live 全体についての主要なテスターで、ポーランド語に関わる多くの貢献（フォントや Windows インストールなど）をしてくれました。Staszek 氏は 2018 年 2 月に亡くなりましたが、私たちは彼のことを忘れません。
- Olaf Weber。過去何年にも渡って辛抱強く Web2C のメンテナンスを行ってくれました。
- Gerben Wierda。初期の macOS サポートの開発とメンテナンスを行ってくれました。
- Graham Williams。TeX カタログの創始者です。
- Joseph Wright。各種エンジンで同じプリミティブを使用できるようにするため多

くの貢献をしてくれました。

- 山下弘展。pTEX と関連する多くのサポートを担ってくれました。

バイナリのビルド担当者：Marc Baudoin (amd64-netbsd, i386-netbsd), Ken Brown (i386-cygwin, x86_64-cygwin), Simon Dales (armhf-linux), 角藤亮 (win32), Dick Koch (universal-darwin), Mojca Miklavec (x86_64-darwinlegacy), Norbert Preining (aarch64-linux, armhf-linux, i386-freebsd, amd64-freebsd, i386-linux, x86_64-linux, i386-solaris, x86_64-solaris, x86_64-linuxmusl)。TEX Live のビルドプロセスについては https://tug.org/texlive/build.html を参照してください。

このマニュアルの翻訳者：朝倉卓人（日本語）、Denis Bitouzé、Patrick Bideault（フランス語）、Carlos Enriquez Figueras（スペイン語）、Jjgod Jiang、Jinsong Zhao、Yue Wang、Helin Gai（中国語）、Nikola Lečić（セルビア語）、Marco Pallante、Carla Maggi（イタリア語）、Petr Sojka、Jan Busa（チェコ・スロバキア語）、Boris Veytsman、Vladimir Lomov（ロシア語）、Uwe Ziegenhagen（ドイツ語）。TEX Live のドキュメントウェブページは https://tug.org/texlive/doc.html です。

もちろん、最も重要な謝辞は Donald Knuth に対して贈られるべきです。まず TEX を発明してくれたこと、そしてそれを世界に広めてくれたことについて、深く感謝します。

9 更新履歴

9.1 過去

最初に議論が始まったのはオランダ TEX ユーザ会が 4AllTEX CD を MS-DOS ユーザのために用意し始めたときのことでした。そのときは、すべてのシステムに向けて 1 つの合理的な CD を作成することが望ましいとされました。これは、当時あまりに漠然とした目標でしたが、結果的には 4AllTEX CD を完成させたのみならず TEX ディレクトリ構造（TDS, https://tug.org/tds）についての作業チームを TUG に発足させることにつながりました。作業チームはどのようにすれば一貫性があり、管理の容易な TEX サポートファイルのコレクションを作成できるか議論を重ねました。TDS についての完全な原稿は 1995 年 12 月発行の *TUGboat* に掲載されましたが、かなり早い段階から望ましい成果物は CD 上のモデル構造であることは明らかでした。いま皆さんにとってお馴染みのディストリビューションは、この作業チームの議論のかなり直接的な成果と言えます。また 4AllTEX CD の成功から、同様の簡単なシステムを作れば Unix 系システムのユーザも恩恵に与ることができることも明らかで、この事実も TEX Live にもう 1 つの大きな流

れを作りました。

私たちは、1995 年の秋にまず Unix 系の TDS CD の開発に取り掛かり、すぐに Thomas Esser の teTEX が理想的なディストリビューションであることがわかりました。teTEX はすでに複数のプラットフォームをサポートしており、さまざまなファイルシステムに対応するように設計されていました。Thomas 氏の協力を得て活動は 1996 年のはじめに本格化し、最初のバージョンが 1996 年の 5 月にリリースされました。1997 年の頭に Karl Berry が大規模な修正を加えた新しいバージョンの Web2C をリリースし、Thomas 氏が teTEX に加えていたほとんどすべての機能が利用可能になりました。この Web2C に teTEX の `texconfig` スクリプトを加えたものはバージョン 2 のベースに採用されることが決まりました。バージョン 3 は Olaf Weber によってさらに大きく修正された Web2C 7.2 をベースにしました。同時に、新しいバージョンの teTEX も開発され、TEX Live もすぐにそのほとんどの新機能を取り込みました。バージョン 4 も同じパターンでのリリースとなり、新しい teTEX と新しい Web2C 7.3 を利用して作成されました。Fabrice Popineau の貢献もあり、この時点までに TEX Live は完全な Windows サポートを完成させていました。

バージョン 5（2000 年 3 月）では多くの部分が改修・チェックされ、何百ものパッケージがアップデートされました。パッケージの詳細は XML ファイルに保存されるようになりました。一方で TEX Live 5 における大きな変更として、すべての有償ソフトウェアは取り除かれました。このときから TEX Live に含まれるすべてのものは Debian フリーソフトウェアガイドライン（https://www.debian.org/intro/free）互換であることが求められるようになりました。私たちはすべてのパッケージのライセンスを確認するよう最善を尽くしましたが、もし何か間違いがあればご連絡ください。

バージョン 6（2001 年 7 月）にはさらに多くのパッケージアップデートが取り込まれました。このバージョンで最大の更新点はインストールについての新しい選択肢です。ユーザはより詳細に必要とするコレクションを選択できるようになりました。特定の言語と関連するコレクションが大きく見直され、そうしたコレクションを選択すれば対応するマクロやフォントがインストールされるだけでなく、適切な `language.dat` も用意されるようになりました。

2002 年のバージョン 7 での大きな変更としては Mac OS X サポートの追加と、例年通り無数のパッケージやプログラムのアップデートが挙げられます。このときの大きな目標はソースコードを teTEX のそれと統合し、バージョン 5, 6 間の不整合を修正することでした。

9.1.1 2003

2003 年も例年通り数多くのアップデートと追加パッケージがあり TEX Live は巨大化してもはや 1 枚の CD に収まらないことがわかりました。そのため、私たちはこれを 3 つの異なるディストリビューションに分離しました（2.1 節（p. 9）を参照）。その他の変更点：

- LATEX チームからのリクエストを受けて、標準設定における `latex`, `pdflatex` コマンドが ε-TEX エンジンを使用するように変更（p. 13 参照）
- 新しい Latin Modern フォントを追加（そして推奨）
- Alpha OSF のサポートを終了（HPUX のサポートは既に終了していました）。これは新しいバイナリをコンパイルするためのハードウェアを誰も所有していなかった（そして新しい有志も現れなかった）ためです。
- Windows サポートの大きな変更。XEmacs ベースの統合環境が始めて導入されました。
- Windows 向けの重要な追加プログラム（Perl, Ghostscript, ImageMagic, Ispell）が TEX Live のディレクトリにインストールされるようになりました。
- `dvips`, `dvipdfm`, `pdftex` などによって使用されるフォントマップファイルが新しい `updmap` によって生成され、`texmf/fonts/map` に配置されるようになりました。
- TEX, METAFONT, METAPOST がデフォルトでほとんどの入力文字（文字コード 32 以上）をそのままファイル（例えば `\write` の出力先）、ログファイル、ターミナルに出力するようになりました。すなわち `^^` 表現への変換が行われないようになりました。TEX Live 7 では、この変換はシステムのロケール設定に依存していました。今後はロケール設定は TEX のコアプログラムの挙動に影響を与えなくなります。もし何らかの理由により `^^` 表現による出力が必要な場合は `texmf/web2c/cp8bit.tcx` のファイル名を変更してください。（将来的にはもう少し簡単な方法を用意する予定です。）
- このドキュメントを全面的に改訂しました。
- 最後に、バージョン番号が大きくなってきたので、今後は単にリリース年で表すことにしました。すなわち今回のバージョンは TEX Live 2003 となります。

9.1.2 2004

2004 年には数多くの更新がありました：

- ローカルにインストールしたフォントで、独自の .map や（あまりないと思いますが）.enc を利用するものがある場合、それらのサポートファイルを移動する必要があります。
  .map ファイルについては TEXFONTMAPS パスに含まれる（各 TEXMF ツリーの）fonts/map 以下のサブディレクトリのみが検索されるようになりました。同様に .enc ファイルについても ENCFONTS パスに含まれる fonts/enc 以下のサブディレクトリのみが検索対象になりました。しばらくの間は updmap コマンドが将来的に問題となり得るファイルについて警告を出します。
  より詳細な対策その他の情報については https://tug.org/texlive/mapenc.html を参照してください。
- TEX コレクションに MiKTEX ベースのインストール CD が追加されました。MiKTEX 実装の Web2C がお好みの場合はそちらをお使いください。詳細は 2 節（p. 9）を参照。
- 従前のリリースにおいて、TEX Live における唯一の巨大な TEXMF ツリーであった texmf を texmf, texmf-dist, texmf-doc の 3 つに分割しました。詳細は 2.2 節（p. 10）と各ツリーの README を参照してください。
- すべての TEX 関連の入力ファイルを各 texmf* ツリーの tex サブディレクトリに集約しました（これまでは tex, etex, pdftex, pdfetex 等々に分けられていました）。詳細は texmf-dist/doc/generic/tds/tds.html#Extensions を参照してください。
- （ユーザによって直接実行するためのものではない）補助スクリプトを各 texmf* ツリーの scripts サブディレクトリに移動し、kpsewhich --format=texmfscripts で探せるようにしました。こうした補助スクリプトを呼び出すプログラムについては改修の必要があります。詳細は https://tug.org/tds/tds.html#Scripts を参照してください。
- cp227.tcx の設定により、ほとんどすべてのフォーマットで大半の文字を ^^ 表現へ変換することなくそのまま出力するようになりました。特に文字コード 32–256 の文字とタブ、垂直タブ、改ページは印字可能文字として扱い変換されません。例外は plain TEX（32–126 のみ印字可能）、ConTEXt（0–256 を印字可能）と Ω 関連フォーマットです。デフォルトの挙動は TEX Live 2003 とほとんど変わりませんが、実装が整理されカスタマイズも容易になりました。詳細は texmf-dist/doc/web2c/web2c.html#TCX-files を確認してください。（ところで Unicode 入力については TEX はエラーを表示して部分的な文字シークエンス

を出力する可能性があります。これは TEX がバイト指向であるためです。)

- pdfetex が (plain) tex を除くすべてのフォーマットのデフォルトエンジンになりました（もちろん、latex 等で実行された場合は DVI を出力します)。つまり、ε-TEX 拡張（texmf-dist/doc/etex/base/）に加えて pdfTEX の microtypography 機能も LATEX や ConTEXt で利用可能になります。
- この変更は ifpdf（plain TEX でも LATEX でも使用可）または同等のコードの利用がこれまで以上に重要となったということを意味します。なぜなら単純に \pdfoutput（または他のプリミティブ）が定義されているかどうかを調べるだけでは PDF 出力が行われているかどうかを判定できなくなったからです。この後方互換性について今年のリリースでは可能な限り対処しましたが、来年以降は DVI 出力を行っているときであっても \pdfoutput が定義されている可能性があります。
- pdfTEX（https://tug.org/applications/pdftex/）には多くの新機能が追加されました：
  - \pdfmapfile と \pdfmapline がドキュメント内からのフォントマップアクセスをサポートします。
  - Microtypographic フォント展開がより簡単に利用できるようになりました。https://www.ntg.nl/pipermail/ntg-pdftex/2004-May/000504.html
  - 従来は専用の設定ファイル pdftex.cfg で設定していたパラメタは、今後プリミティブを用いて設定されなければなりません。具体的には pdftexconfig.tex で設定されています。pdftex.cfg はもはやサポートされません。pdftexconfig.tex が変更された場合、すべての .fmt ファイルを再生成する必要があります。
  - その他の更新点については pdfTEX のマニュアルを参照してください：texmf-dist/doc/pdftex/manual/pdftex-a.pdf
- tex（および mf と mpost）の \input プリミティブがダブルクオートで囲うことによって空白やその他の特殊文字を含む入力を受け付けるようになりました。典型的な使用例：

```
\input "filename with spaces"   % plain
\input{"filename with spaces"}  % latex
```

詳細については Web2C のマニュアルを参照してください texmf-dist/doc/web2c。

- Web2C に encTEX サポートが追加され、この機能は結果的にすべての TEX プログラムで -enc オプションを通して利用可能になりました（**ただしフォーマットが生成されている場合に限る**）。encTEX は入出力の再エンコーディングについて汎用的なサポートを提供するもので、Unicode（UTF-8 形式による）のフルサポートを可能としています。詳細については texmf-dist/doc/generic/enctex/ と http://olsak.net/enctex.html を参照してください。
- ε-TEX と Ω を組み合わせた新エンジン ℵ が利用可能になりました。texmf-dist/doc/aleph/base と https://texfaq.org/FAQ-enginedev にいくらかの情報があります。ℵ における LATEX ベースのフォーマットは lamed といいます。
- 最新の LATEX は新しいバージョンの LPPL の下でインストールされました。これで公式に Debian 公認のライセンスになりました。他のアップデートについての情報と合わせて texmf-dist/doc/latex/base の ltnews ファイルに詳細が記載されています。
- DVI を PNG 画像ファイルに変換する新しいプログラム dvipng が追加されました。https://ctan.org/pkg/dvipng を参照。
- medium サイズのフォント向けに cbgreek パッケージを導入しました。これはフォント作者（Claudio Beccari）の同意と助言に基づくものです。除外されているフォントは不可視・アウトライン・透過文字です。これらは比較的使用頻度の低いもので、私たちは容量を気にする必要がありました。フルセットのフォントはもちろん CTAN から入手できます（http://mirror.ctan.org/tex-archive/fonts/greek/cbfonts）。
- oxdvi は廃止されました。xdvi を使用してください。
- initex など tex, mf, mpost の ini, vir 版コマンド（リンク）は作成されなくなりました。ini の機能は何年も前からコマンドラインオプション -ini を通して利用可能です。
- i386-openbsd プラットフォームのサポートは廃止されました。BSD ポートシステム向けの tetex パッケージと GNU/Linux および FreeBSD バイナリが利用可能なので、ビルド担当のボランティアは他のことに時間と労力を割いた方が良いようです。
- （少なくとも）sparc-solaris においては t1utils を実行する際には環境変数 LD_LIBRARY_PATH を設定する必要があります。なぜなら、このプログラムは C++ によってコンパイルされていますが、それらについて実行時ライブラリの標準的な配置位置が存在しないためです（これは今に始まったことではありません

が、これまでドキュメント化されていませんでした)。同様に mips-irix においても、MIPSpro 7.4 ランタイムが必要です。

9.1.3 2005

2005 年も例年通り膨大なパッケージとプログラムのアップデートがありました。前年と比べるとインフラは安定していますが、必然的にいくつかの変更はありました：

- システムツリーの設定を変更する新しいスクリプト texconfig-sys, updmap-sys, fmtutil-sys が導入されました。texconfig, updmap, fmtutil スクリプトは $HOME/.texlive2005 以下のユーザ設定を変更するようになりました。
- 上の変更に対応して、設定ファイルを探すべき TEXMF ツリーを指定する新しい環境変数 TEXMFCONFIG と TEXMFSYSCONFIG（それぞれユーザ用とシステム用）が追加されました。したがって個人用の fmtutil.cnf や updmap.cfg がある場合は、これらのパス内に移動する必要があります。もしくは texmf.cnf を用いて TEXMFCONFIG, TEXMFSYSCONFIG の値を再定義する手もあります。いずれにしても、実際の設定ファイル位置と TEXMFCONFIG, TEXMFSYSCONFIG の指す位置は必ず一致していなければなりません。詳細は 2.3 節（p. 11）を参照してください。
- 昨年は pdfTEX が使用されていたとしても、DVI 出力を行う場合には \pdfoutput 等のプリミティブが未定義になるよう後方互換性を維持していました。今年は、予告していたとおり、この後方互換性に対するケアは終了します。そのためもし \ifx\pdfoutput\undefined のようなコードを利用して PDF 出力か否かを判定している場合、修正が必要になります。この場合 ifpdf パッケージ (plain TEX でも LATEX でも使用可能です) を使用するか、そのしくみを真似するのがいいでしょう。
- 昨年、ほとんどのフォーマットで出力文字 (8-bit) がそのまま維持されるようになりました（前のセクションを参照してください)。新しい TCX ファイル empty.tcx を用いてより簡単に ^^ 表現を利用するかつての挙動に戻せるようになりました：

```
latex --translate-file=empty.tcx yourfile.tex
```

- DVI を PDF に変換するため、新しく dvipdfmx プログラムが追加されました。これは活発にメンテナンスされている dvipdfm の後継です（dvipdfm は今でも利用可能ですが、推奨はされません)。
- Adobe Acrobat Reader を再起動することなく PDF のリロードを可能にする pdfopen と pdfclose が追加されました (xpdf, gv, gsview など Acrobat 以外の PDF

リーダにはそもそもこのような問題はありませんでした)。

- 一貫性のため HOMETEXMF と VARTEXMF はそれぞれ TEXMFHOME と TEXMFSYSVAR に変更されました。ユーザごとのツリーを示す TEXMFVAR もデフォルトで存在していることに注意してください。このリストの最初の項目も再度確認してください。

9.1.4 2006–2007

2006–2007 年における TEX Live への大きな出来事は XETEX の追加でした。このプログラムは xetex と xelatex として利用可能です（https://scripts.sil.org/xetex 参照)。

pdfTEX（https://tug.org/applications/pdftex）と同様に、METAPOST にも将来的な拡張の計画も含めて特筆すべきアップデートがありました https://tug.org/metapost/articles。

TEX の .fmt（高速フォーマット）と METAPOST, METAFONT 用の同等ファイルは texmf/web2c 直下から同サブディレクトリに移されました（一方で texmf/web2c 直下も既存の .fmt ファイルのために検索対象に残っています)。それらのサブディレクトリにはそれぞれのエンジン名（tex, pdftex, xetex 等）が付けられています。通常の使い方をしている限り、この変更による影響はないはずです。

（plain）tex コマンドは先頭行の %&（使用フォーマットの指定に使用する）を解釈しなくなりました。tex コマンドは純粋な Knuthian TEX です。(LATEX など他のコマンドは今後も %& を解釈します。)

当然のことながら、この期間にも（いつも通り）上記以外に何百ものパッケージやプログラムのアップデートがありました。平常通り、それらについては CTAN（https://ctan.org）を確認してください。

内部的には、ソースツリーを Subversion で管理するようになり、その標準 Web インターフェースでソースツリーを閲覧できるようになりました（TEX Live のウェブサイトからリンクされています)。この変更は最終的な配布版には関係ありませんが、この変更によって今後の開発基盤が安定すると期待しています。

最後に、2006 年 5 月に Thomas Esser が teTEX の更新を終了することを発表しました（https://tug.org/tetex)。結果として、特に GNU/Linux のディストリビュータから、にわかに TEX Live が注目されることになりました。TEX Live には新しく tetex スキームが追加され、これを用いると旧来の teTEX と概ね同等のものをインストールすることができます。私たちは、早晩この変化が全世界の人々の TEX 環境の改善につながる

ことを願っています。

9.1.5 2008

今年は TEX Live のインフラが全面的に再設計・再実装されました。インストールについての完全な情報は `tlpkg/texlive.tlpdb` というテキストファイルに保存されるようになりました。インフラ刷新による変化は色々ありますが、これによって一度インストールした TEX Live をインターネットを利用してアップグレードできるようになりました。この機能は MiKTEX には何年も前からあったものです。今後は CTAN で新しいパッケージがリリースされる度にアップデートを行うことができるはずです。

大きな出来事として、新しいエンジン LuaTEX（https://luatex.org）が追加されました。このエンジンは組版にさらなる柔軟性を与えるだけでなく、TEX 文書の内外で利用できる素晴らしいスクリプト言語の処理系を提供してくれます。

Windows と Unix 系プラットフォームのサポートがより統一的になりました。具体的には、TEX Live に同梱されるようになった処理系によって Windows でも Perl スクリプトと Lua スクリプトが利用可能になりました。

新しい tlmgr スクリプトは（5 節）は TEX Live の初期インストール後の管理を行う汎用インターフェースです。このコマンドはパッケージアップデートと継続的なフォーマット・マップファイル・言語ファイル等の再生成を行います（オプションで、ローカルに追加されたものについても行います)。tlmgr が開発されたので、これまでフォーマットやハイフネーション設定ファイルを管理していた texconfig は廃止されました。

また新しい索引作成支援プログラム xindy（http://xindy.sourceforge.net/）がほとんどのプラットフォーム向けに提供されるようになりました。

`kpsewhich` コマンドは与えられたファイルについて、すべてのマッチを返すことができるようになりました（オプション `--all`)。特定のサブディレクトリ内のマッチだけを返すことも可能です（オプション `--subdir`)。

`extractbb` コマンドという形で、dvipdfmx プログラムにバウンディングボックス情報を抽出する機能が追加されました。これにより dvipdfm が持っていたほとんどの機能が dvipdfmx にも実装されたことになります。

`Times-Roman` や `Helvetica` のようなフォントエイリアスは削除されました。パッケージごとにそうしたエイリアスに求める挙動が異なり（例えば、期待されるエンコーディングが異なるなど)、またそれに対する良い解決策も存在しないためです。

ポーランド語向けの platex フォーマットは、まったく別の日本語用プログラム platex と名前が衝突しているため削除されました。代わりに、今後は polski パッケージが主とし

てポーランド語サポートを提供します。

内部的には、アップデートの簡便性のため、WEB の文字列プールファイルをバイナリにコンパイルするようになりました。

最後に、'TEX tuneup of 2008' に述べられている Donald Knuth による変更が今回のリリースに反映されました。詳細は https://tug.org/TUGboat/Articles/tb29-2/tb92knut.pdf を参照してください。

9.1.6 2009

2009 年、LuaTEX の OpenType サポート等を活用するため Lua(LA)TEX のデフォルトの出力フォーマットが PDF に変更されました。新しいコマンド名 dviluatex と dvilualatex を使用すると DVI 出力の LuaTEX を実行できます。LuaTEX のウェブサイトは https://luatex.org です。

Ω（Omega）開発者らとの協議の後、オリジナルの Ω エンジンと Λ（Lambda）フォーマットは削除されることになりました。後継である ℵ（Aleph）と ℸ（Lamed）は Ω の機能を担うため今後も TEX Live に残ります。

Computer Modern を含め、新しいバージョンの AMS Type 1 フォントが追加されました。何年も前に Knuth が METAFONT のソース修正を行って変更されたいくつかのシェイプがついに反映され、またヒンティングも更新されました。Euler フォントも Hermann Zapf によって大幅に再デザインされました（https://tug.org/TUGboat/Articles/tb29-2/tb92hagen-euler.pdf 参照）。いずれについてもメトリクスは変更されていません。AMS フォントのウェブサイトは https://ams.org/tex/amsfonts.html です。

新しい GUI フロントエンド TEXworks が Windows と MacTEX 向けに追加されました。他のプラットフォーム向けのものや詳細情報については TEXworks のウェブサイト（https://tug.org/texworks）を参照してください。TEXworks は macOS 向けの統合開発環境 TEXShop にインスパイアされたクロスプラットフォームなフロントエンドで、使い勝手の良さに重きを置いています。

製図用ツール Asymptote が多くのプラットフォーム向けに追加されました。これはテキストベースのグラフィクス記述言語で、大雑把には METAPOST と似たものですが、高度な 3D サポートその他の機能が利用可能です。ウェブサイトは http://asymptote.sourceforge.io です。

独立した dvipdfm コマンドは dvipdfmx に置き換えられました（ただし特別な互換モードで実行されます）。dvipdfmx は CJK サポートと、dvipdfm の最後のリリースから何年にも渡って蓄積された数多くの修正が含まれています。

新しいプラットフォーム（cygwin, i386-netbsd）向けのバイナリが追加されました。ただし OpenBSD ユーザは標準的なパッケージシステムを用いて TEX をインストールすることが推奨されます。複数のバージョンの OpenBSD で動作するバイナリを作成するのは困難でした。

その他の雑多な変更：

- lzma の安定した代替として xz 圧縮（https://tukaani.org/xz/）を採用しました。
- 後続の文字列が定義済みの変数と一致しない限り '$' をファイル名に使用できるようになりました。
- Kpathsea ライブラリがマルチスレッド化されました（METAPOST で使用されています）。
- TEX Live 全体のビルドが Automake ベースになりました。

最後に過去のバージョンに関することについて告知します。TEX Live のすべてのリリースは ftp://tug.org/historic/systems/texlive から入手できます（CD ラベル等のおまけもあります）。

9.1.7 2010

2010 年、出力 PDF のデフォルトバージョンが 1.5 になり、これまで以上の圧縮が可能になりました。この変更は PDF を出力するすべての TEX エンジンと dvipdfmx について適用されました。LATEX パッケージ pdf14 を使用するか \pdfminorversion=4 を設定すると PDF 1.4 出力に戻すことができます。

pdf(LA)TEX は LATEX の graphics.cfg が読み込まれていて、PDF 出力を行っている場合、epstopdf を用いて**自動的に**インクルードされた Encapsulated PostScript（EPS）ファイルを PDF に変換するようになりました。デフォルトのオプションは既存の PDF ファイルが上書きによって失われてしまうことのないように設定されていますが、\documentclass よりも前に \newcommand{\DoNotLoadEpstopdf}{}（\def を用いてもよい）を記述して読み込み済みのように見せかけることで epstopdf 使用を回避できます。また pst-pdf パッケージを使用した場合も読み込みは行われません。詳細については epstopdf パッケージのドキュメントを参照してください（https://ctan.org/pkg/epstopdf-pkg）。

関連する変更として、今年からいくつかの定められた外部コマンドが TEX から（\write18 機能を介して）デフォルトで実行できるようになりました。デフォルトで実行が許可されているコマンドは repstopdf, makeindex, kpsewhich, bibtex, bibtex8

です。このリストは texmf.cnf に定義されています。こうした外部コマンドをすべて禁止する必要のある環境では、インストーラにオプションを指定するか（3.2.4 節を参照）、tlmgr conf texmf shell_escape 0 を実行して設定を上書きしてください。

上とは別の関連する変更は BibTeX と Makeindex が任意ディレクトリのファイル出力を（TeX 本体と同様）デフォルトでは行わなくなったことです。これは、これらのプログラムを制限付き \write18 で実行できるようにするための措置です。この挙動を変更するためには環境変数 TEXMFOUTPUT を設定するか openout_any 設定を変更してください。

XeTeX が pdfTeX と同様に同じ行内でのマージンカーニングをサポートするようになりました（フォント展開は今のところサポートされていません）。

tlmgr がデフォルトで各パッケージのバックアップを取るようになったので（tlmgr option autobackup 1）、パッケージアップデート時に不具合が発生しても tlmgr restore により簡単に復元できるようになりました。もしインストール後のアップデートを行いたいもののバックアップ用のディスク容量がないという場合は tlmgr option autobackup 0 を実行してください。

新規に追加されたプログラム：

- pTeX エンジンと関連する日本語組版のためのユーティリティ
- BibTeXU: Unicode サポートが有効な BibTeX
- chktex: (LaTeX 文書をチェックするツール（元々 http://baruch.ev-en.org/proj/chktex から提供されていたもの）
- dvisvgm: DVI を SVG に変換するツール（https://dvisvgm.de）

新しいプラットフォーム向けの実行バイナリも追加されました：amd64-freebsd, amd64-kfreebsd, i386-freebsd, i386-kfreebsd, x86_64-darwin, x86_64-solaris.

TeX Live 2009 の変更ですが、昨年リリースノートに書き忘れたものがありました。TeX4ht 関連の多くの実行バイナリ（https://tug.org/tex4ht）がバイナリ用ディレクトリから削除されました。汎用の mk4ht コマンドを用いてすべての tex4ht コンビネーションを実行することができます。

最後に、TeX コレクション DVD 上の TeX Live はライブ実行できなくなりました。1 枚の DVD だけではもはや十分なスペースが確保できないためです。有用な副作用として、物理 DVD からのインストールがこれまでよりかなり高速になりました。

9.1.8 2011

macOS バイナリ（universal-darwin, x86_64-darwin）の動作要件が Leopard 以降になりました。今後 Panther と Tiger はサポート対象外となります。

一般的なプラットフォーム向けに文献処理ツール biber が追加されました。このプログラムの開発は LaTeX の参考文献関連機能の完全な再実装である biblatex パッケージの開発と緊密に連携しています。

METAPOST（mpost）は .mem ファイルを作成したり利用したりしなくなりました。plain.mp のような必要ファイルは実行の度に読み込まれなければなりません。これは METAPOST をライブラリとして利用できるようにしたことに関連する変更です（ライブラリ利用可能になったこともまた顕著な変更ですが、一般のユーザにはあまり関係がありません）。

updmap の Perl 実装はこれまで Windows 専用でしたが、刷新されてすべてのプラットフォームで利用されるようになりました。これもユーザに影響のあるような変更ではないはずですが、実行速度はかなり向上しました。

initex, inimf コマンドが復活しました（ただし、その他の ini* コマンドは復活していません）。

9.1.9 2012

tlmgr が複数のネットワークレポジトリからのアップデートをサポートするようになりました。tlmgr help の出力の “multiple repositories” に関するセクションにはさらなる詳細が記述されています。

xetex, xelatex の双方について \XeTeXdashbreakstate のデフォルト値が 1 になりました。これにより em-ダッシュと en-ダッシュの直後の改行が許容されます（これは plain TeX, LaTeX, LuaTeX 等の挙動と同じです）。既存の XeTeX 文書で、これまでの行分割とまったく同じ挙動を必要とする場合は明示的に \XeTeXdashbreakstate を 0 に設定してください。

pdftex と dvips を含むいくつかのプログラムが生成するファイルが 2 GB を超えられるようになりました。

あまりにも多くのバリエーションが存在するため、35 の PostScript 標準フォントが dvips の出力にデフォルトで埋め込まれるようになりました。

制限付き \write18 実行モードにおいて、mpost がデフォルトで許可されるようになりました。

texmf.cnf が ../texmf-local の位置（例えば /usr/local/texlive/texmf-local/web2c/texmf.cnf）でも（存在すれば）みつけられるようになりました。

updmap スクリプトがグローバルな設定ではなくツリー毎に updmap.cfg を読み込むよ

うになりました。この変更は、updmap.cfg を直接変更していない限りユーザへの影響はないはずです。詳細は updmap --help の出力を参照してください。

プラットフォーム：armel-linux と mipsel-linux が追加され、sparc-linux と i386-netbsd はもはやメインディストリビューションには含まないことになりました。

9.1.10 2013

ディストリビューション構造：トップレベルの texmf ディレクトリは、簡潔のため texmf-dist と統合されました。Kpathsea 変数 TEXMFMAIN と TEXMFDIST は今後ともに texmf-dist を参照することになります。

多くの小さな言語サポートコレクションは、インストールをシンプルにするため、統合されました。

METAPOST: PNG 出力のネイティブサポートと浮動小数点数（IEEE double）が追加されました。

LuaTEX: 内蔵 Lua 処理系のバージョンが 5.2 に上がりました。また、外部 PDF のページコンテンツを処理など多くの機能をもつ pdfscanner ライブラリが新たに追加されました（詳細は pdfscanner のウェブサイトを参照してください）。

XETEX（ウェブサイトにさらに多くの情報があります）：

- フォントレイアウトに ICU に代わって HarfBuzz ライブラリを使用するようになりました（ICU も入力エンコーディング、双方向組版、オプショナルな Unicode 改行をサポートするために依然として利用されてはいます）。
- グラファイトレイアウトには SilGraphite に代わって Graphite2 と HarfBuzz が利用されるようになりました。
- Mac では ATSUI（廃止予定）に代わって Core Text を使用するようになりました。
- 同名のものがある場合、Type1 フォントより TrueType/OpenType フォントが優先されるようになりました。
- 稀に XETEX と xdvipdfmx でフォント検索の挙動に違いが生じる問題を修正しました。
- OpenType の math cut-ins をサポートしました。

xdvi: レンダリングに t1lib に代わって FreeType を使用するようになりました。

microtype: XETEX の protrusion 機能と LuaTEX の protrusion、フォント展開、トラッキング機能をサポートしたほか、数々の拡張が行われました。

tlmgr: 新しく pinning アクションが追加され、複数のレポジトリの管理が簡単になりました。tlmgr --help 出力の該当セクションまたはオンライン https://tug.org/

texlive/doc/tlmgr.html#MULTIPLE-REPOSITORIES に詳細が記述されています。

プラットフォーム：armhf-linux, mips-irix, i386-netbsd, amd64-netbsd が追加（または復活）されました。一方 powerpc-aix は廃止されました。

9.1.11 2014

2014 年も Knuth による TEX のアップデートがありました。この変更はすべてのエンジンに影響しますが、ユーザから見える変化は起動時バナーの `preloaded format` の文字列ぐらいでしょう。Knuth によれば、今回の変更により起動時バナーはバイナリに実際にアンダンプされて読み込まれたフォーマットよりも、読み込まれていることが**期待される**フォーマットを反映するようになりました。バナーの内容はさまざまな方法で上書きされ得ます。

pdfTEX: 警告を抑制する新しいパラメタ `\pdfsuppresswarningpagegroup` と、PDF テキストリフローを補助するために単語内スペースを制御する新しいプリミティブ `\pdfinterwordspaceon`, `\pdfinterwordspaceoff`, `\pdffakespace` が追加されました。

LuaTEX: フォント読み込みとハイフネーションについて、大きな変更と修正がありました。最大の新要素は新しい TEX エンジンで、`luajittex` とその仲間である `texluajit`, `texluajitc` が追加されました。これらは Lua の JIT コンパイラを利用しています（詳細は *TUGboat* の記事 https://tug.org/TUGboat/tb34-1/tb106scarso.pdf にあります）。`luajittex` は現在開発途上にあり、すべてのプラットフォームで利用可能なわけではなく、また `luatex` ほど安定していない可能性があります。TEX Live チームも `luajittex` 開発陣も、このエンジンは Lua コードの JIT コンパイルについて実験を行う場合を除いては、使用を推奨していません。

XETEX: Mac を含むすべてのプラットフォームで同じ画像フォーマットをサポートするようになりました。Unicode の互換性分解フォールバックは回避されるようになりました（ただしそれ以外については未対応）。XETEX の以前のバージョンとの互換性のため、グラファイトフォントについては OpenType を優先します。

METAPOST: `numberprecision` に加えて、新しい数値体系 `decimal` のサポートされるようになりました。Knuth により `plain.mp` の `drawdot` の定義が更新されました。SVG および PNG 出力についてのバグが修正されました。

TEX Live 2014 リリース後どこかのタイミングで ConTEXt のユーティリティ `pstopdf` は独立のコマンドとしては廃止される予定です。これは同名の OS ユーティリティとの衝突があるためです。同ユーティリティは `mtxrun --script pstopdf` によりこれからも

呼び出すことができます。

psutils は新しいメンテナにより大きく修正されました。結果として、いくつかのあまり使われていないユーティリティ（fix*, getafm, psmerge, showchar）はユーザレベルバイナリとして提供するのではなく scripts ディレクトリ以下に移動されることになりました（もしこの変更で問題が発生した場合は戻されるかもしれません）。一方で新しいスクリプト psjoin が追加されました。

TEX Live ベースのディストリビューション MacTEX (3.1.2 節) は、今後 Latin Modern と TEX Gyre フォントについての Mac 専用パッケージを同梱しないことになりました。これはユーザが各自でそれらのフォントをシステムで利用できるように設定するのはそれほど難しくないためです。また TEX4ht（とりわけ tex4ht.env）が Ghostscript を直接利用するようになったので ImageMagick の convert コマンドも削除されました。

中国語、日本語、韓国語をサポートする langcjk コレクションは大きすぎたので言語ごとに独立のパッケージに分割されました。

プラットフォーム：x86_64-cygwin が追加された一方、mips-irix は削除されました。Microsoft が Windows XP のサポートを終了したので、TEX Live も XP ではそのうち動作しなくなるでしょう。

9.1.12 2015

LATEX 2_ε に元々は fixltx2e パッケージを読み込まない限りは適用されなかった変更がデフォルトで組み込まれるようになりました（逆に、今後は fixltx2e を読み込んでも何も起こりません）。新たに追加された latexrelease パッケージその他の新機構を用いて「どこまでの変更を反映するか」をコントロールすることができます。詳細は付属ドキュメントの “LATEX News #22” や “LATEX changes” に綴られています。ちなみに babel と psnfss は LATEX の中核を成すパッケージながら開発・保守は独立に行われており、今回の変更とは関係がありません（これまで通り動作するはずです）。

内部的には LATEX 2_ε 自身が Unicode 関連のエンジン設定（カテゴリーコードの初期値など）を同梱するようになりました。こうした設定は以前は TEX Live が提供していました。この変更はユーザには無関係のはずです。低レベルの内部コントロールシークエンスはリネームされたり削除されたりしていますが、全体としての挙動はこれまでと完全に一致しているはずです。

pdfTEX: JFIF と同様 JPEG の Exif 情報もサポートするようになりました。\pdfinclusionerrorlevel の値が負のときは警告を一切発しません。また xpdf 3.04 に対応しました。

LuaTEX: トークン読み取りのための newtokenlib ライブラリが追加されました。normal な乱数生成やその他の部分に関わるバグ修正も含まれています。

XETEX: 画像の取り扱いに関する修正がありました。xdvipdfmx バイナリは最初に xetex の仲間を検索するようになりました。XDV の内部オペコードが変更されました。

METAPOST: 新しい数値体系 binary が追加されました。up*tex に対応する日本語用の upmpost, updvitomp プログラムが追加されました。

MacTEX: CJK サポート Ghostscript パッケージの更新が反映されました。システム環境設定の "TEX Distribution" ペインが OS X 10.10 Yosemite でも利用可能になりました。リソースフォークフォント（通常拡張子なし）はもはや XETEX ではサポートされなくなりました。一方データフォークフォント（.dfont）は今後もサポートされます。

インフラ：fmtutil スクリプトが updmap と同様、ツリー毎に fmtutil.cnf を読み込むように再実装されました。Web2C の mktex* スクリプト（mktexlsr, mktextfm, mktexpk など）は毎回 PATH にあるプログラムよりも、それぞれと同じディレクトリにあるプログラムを優先するようになりました。

プラットフォーム：kfreebsd 系のプラットフォームサポートは終了しました。これらのプラットフォームでは、それぞれシステムネイティブな機能を利用して簡単に TEX Live を導入できるようになったためです。いくつかの追加プラットフォームがカスタムバイナリの形でサポートされるようになりました（https://tug.org/texlive/custom-bin.html）。加えていくつかのプラットフォームサポートが DVD では省略されるようになりましたが（単純に容量不足のためです）、インターネット経由でこれまで通りインストールすることができます。

9.1.13 2016

LuaTEX: プリミティブについての大きな変更（リネームと削除）とノード構造の部分的な再設計がありました。この変更は Hans Hagen 氏による記事 "LuaTEX 0.90 backend changes for PDF and more"（https://tug.org/TUGboat/tb37-1/tb115hagen-pdf.pdf）に要約されています。すべての詳細については LuaTEX のマニュアル texmf-dist/doc/luatex/base/luatex.pdf を参照してください。

METAFONT: METAFONT に Lua 処理系を組み込んだ実験的なプログラム MFlua と MFluajit が追加されました。現状これらは試用と実験のためのものです。

METAPOST: バグ修正と METAPOST 2.0 に向けた内部的準備が行われました。

LuaTEX とオリジナルの TEX を除くすべての処理系が SOURCE_DATE_EPOCH に対応しました（LuaTEX も次のリリース時に対応する予定です）。環境変数 SOURCE_DATE_EPOCH

が設定されている場合、その値が PDF 出力のタイムスタンプとして利用されます。SOURCE_DATE_EPOCH_TEX_PRIMITIVES も設定されている場合、SOURCE_DATE_EPOCH の値が TEX のプリミティブ \year, \month, \day, \time の初期化にも利用されます。pdfTEX のマニュアルに詳細と使用例が記載されています。

pdfTEX: 通常実行の毎に異なる値の出力をコントロールするためのプリミティブ \pdfinfoomitdate, \pdftrailerid, \pdfsuppressptexinfo が追加されました。これらの機能は PDF 出力のためのもので、DVI のものではありません。

XETEX: プリミティブ \XeTeXhyphenatablelength, \XeTeXgenerateactualtext, \XeTeXinterwordspaceshaping, \mdfivesum が追加されました。文字クラスの上限が 4096 に制限されました。DVI の id バイトが更新されました。

その他のユーティリティ：

- gregorio はグレゴリオ聖歌楽譜を組版するためのパッケージ gregoriotex に含まれる新しいプログラムです。このツールは shell_escape_commands にデフォルトで含まれています。
- upmendex は索引作成プログラムで、概ね makeindex と互換なものですが、Unicode ソートをサポートする他、いくつかの変更が加えられています。
- afm2tfm はアクセントベースの高さ調整を上向きのみに行うようになりました。-a を使用するとすべての調整を省略します。
- ps2pk が拡張 PK/GF フォントを扱えるようになりました。

MacTEX: システム環境設定の “TEX Distribution” ペインが廃止され、そこで取り扱われていた機能は TEX Live ユーティリティに移されました。同梱の GUI アプリケーションはアップデートされました。さまざまな CJK フォントを Ghostscript で扱いたいユーザのために新しく cjk-gs-integrate スクリプトが追加されました。

インフラ：システムレベルの tlmgr 設定ファイルがサポートされました。またパッケージのチェックサムを認証する機能が追加されました。GPG が利用可能な場合はネットワークアップデート時に署名も認証されるようになります。これらの認証はインストーラと tlmgr の両方で行われます。GPG が利用できない場合、アップデートはこれまで通り処理されます。

プラットフォーム：alpha-linux と mipsel-linux は削除されました。

9.1.14 2017

LuaTEX: コールバックや組版制御の機能が追加され、これまで以上に内部処理にアクセスできるようになりました。動的なコード読み込みを行うための ffi ライブラリがい

くつかのプラットフォーム向けに追加されました。

pdfTEX: 昨年導入された環境変数 SOURCE_DATE_EPOCH_TEX_PRIMITIVES は FORCE_SOURCE_DATE にリネームされました（機能はまったく同一です）。トークンリスト \pdfpageattr が文字列 /MediaBox を含んでいる場合、デフォルト /MediaBox の出力が省略されるようになりました。

XETEX: Unicode/OpenType の math 機能は HarfBuzz の MATH テーブルサポートに基づくようになりました。またいくつかのバグが修正されました。

dvips: dvipdfmx との一貫性と他のパッケージから期待される挙動に合わせるため、最後の用紙サイズ special が適用されるようになりました。-L0 オプションを使用（または設定ファイルで L0 指定）した場合、最初の special が優先される従来の挙動に戻ります。

ε-pTEX, ε-upTEX: pdfTEX 由来のプリミティブ \pdfuniformdeviate, \pdfnormaldeviate, \pdfrandomseed, \pdfsetrandomseed, \pdfelapsedtime, \pdfresettimer が追加されました。

MacTEX: 今年から、MacTEX としては Apple がセキュリティパッチを提供している macOS のバージョンのみをプラットフォーム x86_64-darwin としてサポートすることになりました。これにより、現在サポート対象となるのは Yosemite, El Capitan, Sierra（つまり macOS 10.10 以上）です。それよりも古いバージョンの macOS 向けのバイナリは MacTEX には同梱されていませんが、TEX Live には含まれています（x86_64-darwinlegacy, i386-darwin, powerpc-darwin）。

インフラ：TEXMFLOCAL がデフォルトで TEXMFSYSCONFIG および TEXMFSYSVAR に先んじて検索されるようになりました。これはローカルなファイルがシステムのそれを上書きするのがより期待される挙動であろうとの判断からです。また tlmgr に新しく shell モード（対話的実行もしくはスクリプトから使用されることを意図するものです）と conf auxtrees アクション（独自ツリーの追加と削除が容易にできます）が追加されました。

updmap と fmtutil: これらのスクリプトは明示的にいわゆるシステムモード（updmap-sys, fmtutil-sys コマンドの実行または -sys オプションの利用）またはユーザモード（updmap-user, fmtutil-user コマンドの実行または -user オプションの利用）を指定せずに実行された場合、警告を表示するようになりました。これは長年に渡って意図せずユーザモードでスクリプトが実行され、その後システム側のアップデートが反映されない問題へ対処するための措置です。詳細は https://tug.org/texlive/scripts-sys-user.html を参照してください。

install-tl: macOS における TEXMFHOME 等の個人用パスのデフォルト値が MacTEX のデフォルト値 ~/Library/... に合わせられました。新しく -init-from-profile オ

プションが追加され、指定したプロフィールの値に基づいてインストールを開始できるようになったほか、明示的にプロフィールを保存する P コマンドが追加されました。またプロフィールで指定する変数の名前が更新されました（古い変数名も引き続き使用可能です）。

SyncTEX: 一時ファイルの名前が foo.synctex.gz(busy) のような形から foo.synctex(busy) に変更されました（.gz の部分が省略されます）。一時ファイルを削除するフロントエンドやビルドシステムは対応が必要かもしれません。

その他のユーティリティ：texosquery-jre8 は新しいクロスプラットフォームなプログラムで、TEX 文書の中からロケールその他の OS 情報を収集するためのものです。このツールは制限付きシェル実行時でも起動できるよう、デフォルトで shell_escape_commands に追加されています（texosquery は JRE の古いバージョンもサポートしていますが、古い JRE で実行する場合には制限モードでは実行できません。古い JRE は既に Oracle によってサポートされておらず、セキュリティ上の懸念があるためです）。

プラットフォーム：上の MacTEX の項目を確認してください。それ以外の変更はありません。

9.1.15 2018

Kpathsea: システムディレクトリ以外に対してはデフォルトで大文字・小文字を区別せずファイル検索を行うようになりました。これまで通り大文字・小文字を区別したい場合は texmf.cnf または環境変数で texmf_casefold_search を 0 に設定してください。詳細は Kpathsea のマニュアル（https://tug.org/kpathsea）を参照してください。

ε-pTEX, ε-upTEX: 新プリミティブ \epTeXversion が追加されました。

LuaTEX: TEX Live 2019 で Lua 5.3 に移行するための準備として、ほとんどのプラットフォーム向けにバイナリ luatex53 が追加されました。有効化するためには luatex にリネームしてください。もしくは ConTEXt Garden（https://wiki.contextgarden.net）のファイルを使用してください（詳細はサイトを参照してください）。

METAPOST: 間違ったパス方向と TFM, PNG 出力に関する問題が修正されました。

pdfTEX: ビットマップフォントについてエンコーディングベクタが許容されるようになり、カレントディレクトリが PDF ID 中にハッシュされなくなりました。\pdfprimitive とそれに関連するバグ修正が行われました。

XETEX: PDF 画像挿入時に /Rotate がサポートされるようになりました。出力ドライバが正常終了しなかった際の終了ステータスが非ゼロ値になりました。UTF-8 やプリミ

ティブ周りのさまざまな不具合が修正されました。

MacTEX: 下記のサポート macOS バージョン変更を確認してください。加えて、MacTEX によって /Applications/TeX/ 以下にインストールされていたファイルが整理されました。今後このディレクトリ直下には GUI プログラム（BibDesk, LaTeXiT, TEX Live Utility, TEXShop）が置かれ、サブディレクトリに追加のユーティリティやドキュメントが配置されます。

tlmgr: 新しいフロントエンド tlshell（Tcl/Tk）と tlcockpit （Java）が追加されました。それに伴い、tlmgr が JSON 出力に対応しました。uninstall コマンドは remove の別名になりました。新しいアクションおよび同名オプションとして print-platform-info が追加されました。

プラットフォーム：

- armel-linux と powerpc-linux は削除されました。
- x86_64-darwin のサポート対象は macOS 10.10–10.13（Yosemite, El Capitan, Sierra, High Sierra）になりました。
- x86_64-darwinlegacy のサポート対象は 10.6–10.10 です（10.10 については x86_64-darwin の利用をおすすめします）。powerpc-darwin と i386-darwin が削除されたことによって、10.5（Leopard）のサポートは完全的に終了しました。
- Windows XP はもはやサポートされません。

9.1.16 2019

Kpathsea: ブレース展開とパス分割の一貫性が高まりました。設定ファイルでは '.' をハードコードする代わりに新しい変数 TEXMFDOTDIR を使用することになりました。これは、追加のディレクトリやサブディレクトリの検索を容易にするためです（詳細は texmf.cnf 内のコメントを参照してください）。

ε-pTEX, ε-upTEX: 新プリミティブ \readpapersizespecial と \expanded が追加されました。

LuaTEX: Lua のバージョンが新しい算術演算とインターフェースの変更を含む 5.3 に上がりました。PDF 読み込みに独自ライブラリ pplib が用いられるようになったので、poppler への依存がなくなりました（同時に C++ にも依存しなくなりました）。これに伴い Lua インターフェースも変更になっています。

METAPOST: 常に --restricted が有効となる r-mpost コマンドが導入され、同コマンドはデフォルトで制限リストに追加されるようになりました。デシマル・バイナリモードにおける最小精度は 2 になりました。バイナリモードは MPlib ではサポートされなく

なりましたが、スタンドアロンな METAPOST では今でも利用可能です。

pdfTEX: 新プリミティブ \expanded が追加されました。新しいパラメタ \pdfomitcharset が 1 に設定されている場合、/CharSet は PDF 出力で省略されるようになりました。これはその値が PDF/A-2 および PDF/A-3 の仕様を満たすことを保証できないためです。

XETEX: クロスエンジン・サポートのための新プリミティブ \expanded, \creationdate, \elapsedtime, \filedump, \filemoddate, \filesize, \resettimer, \normaldeviate, \uniformdeviate, \randomseed が追加されました。また \Ucharcat はアクティブ文字を生成できるように拡張されました。

tlmgr: ダウンロード用ツール curl をサポートしました。ローカルバックアップの際に（利用可能であれば）lz4 と gzip を xz を適用する前に使用するようになりました。環境変数 TEXLIVE_PREFER_OWN が設定されていない限り、圧縮およびダウンロード用に用いるプログラムについては TEX Live が同梱するものよりもシステムに用意されているものを優先して使用するようになりました。

install-tl: macOS および Windows では新オプション -gui（引数なし）がデフォルトになりました。この状態では新しい Tcl/TK GUI が起動されます（詳細は 1.3 節、3.1.6 節を参照してください）。

ユーティリティ：

- cwebbin（https://ctan.org/pkg/cwebbin）が TEX Live における CWEB 実装となりました。この実装ではより多くの言語がサポートされ、簡易な索引を作成する ctwill プログラムが追加されています。
- chkdvifont: DVI ファイルや tfm/ofm, vf, gf, pk からフォント情報を抽出して報告します。
- dvispc: スペシャルを考慮して、DVI をページ独立にします。

MacTEX: x86_64-darwin のサポート対象は 10.12 以上（Sierra, High Sierra, Mojave）になりました。x86_64-darwinlegacy は引き続き 10.6 以降をサポートします。スペルチェッカ Excalibur はもはや同梱されません（32-bit サポートが必要なためです）。

プラットフォーム：sparc-solaris は削除されました。

9.1.17 2020

全般：

- すべての TEX エンジン（tex を含む）の \input プリミティブが、システム依存の拡張としてファイル名をグループ区切りの引数として受け取れるようになりまし

た。標準的なスペーストークン区切りのファイル名に関する挙動は従来から一切変更されていません。このグループ区切り引数は LuaTEX では元々実装されていたもので、今年からすべてのエンジンで利用可能になりました。ASCII のダブルクオテーション文字（"）はファイル名から除去されますが、その他の変更が引数に対してトークン化後に行われることはありません。この変更は今のところ LaTeX の \input コマンドには影響を与えません。このコマンドは標準の \input プリミティブを再定義したマクロだからです。

- 新しいオプション --cnf-line が kpsewhich, tex, mf とその他すべてのエンジンで利用可能になりました。このオプションは任意の texmf 設定項目をコマンドラインで指定するためのものです。
- ここ数年でさまざまなプリミティブが各種エンジンに追加されたのは、すべての処理系で共通の機能を利用可能にするためです。（cf. *LaTeX News #31*, https://latex-project.org/news）

ε-pTEX, ε-upTEX: 新しいプリミティブ \Uchar, \Ucharcat, \current(x)spacingmode, \ifincsname が主にクロスエンジン・サポートのため追加されました。また \fontchar?? と \iffontchar の仕様が改訂されました。ε-upTEX に限り \currentcjktoken も利用可能になりました。

LuaTEX: HarfBuzz ライブラリが導入され、luahbtex と luajithbtex が追加されました。前者は lualatex で使用されることになりました。新しいプリミティブとして \eTeXgluestretchorder, \eTeXglueshrinkorder が追加されました。

pdfTEX: 新プリミティブ \pdfmajorversion が追加されました。これは単に PDF 出力のバージョン番号を変更するもので、PDF の内容そのものに対しては一切の効果を持ちません。\pdfximage と類似するコマンドは \openin と同様の方法で画像ファイルを検索するようになりました。

pTEX: 新プリミティブ \ifjfont, \iftfont が追加されました。ε-pTEX, upTEX, ε-upTEX も同様です。

XeTEX: \Umathchardef, \XeTeXinterchartoks, \pdfsavepos の挙動が修正されました。

Dvips: ビットマップフォントについてエンコーディングが出力されるようになり、コピー＆ペーストがしやすくなりました（https://tug.org/TUGboat/tb40-2/tb125rokicki-type3search.pdf）。

MacTEX: MacTEX と x86_64-darwin のサポート対象が macOS 10.13 以上（High Sierra, Mojave, Catalina）になりました。x86_64-darwinlegacy は macOS 10.6

以上をサポートします。Apple 社の要請に基づき、MacTEX は公証を受け、そのコマンドラインプログラムは hardened runtime をもつようになっています。BibDesk と TEX Live ユーティリティは公証を受けていないため MacTEX に含まれていませんが、`README` にはそれらを入手できる URL が記載されています。

x86_64-darwinlegacy: `xdvi` を含め、X11 サポートを終了しました。

`tlmgr` とインフラ：

- ダウンロード失敗時に（一度だけ）自動的に再試行するようになりました。
- 新オプション `tlmgr check texmfdbs` が追加されました。これにより `ls-R` ファイルと各ツリーにおける `!!` 指定の整合性を確認することができます。
- パッケージコンテナに `tlnet/archive/`*`pkgname`*`.rNNN.tar.xz` といった形でバージョン情報を含むファイル名が用いられるようになりました。この変更はユーザに直接影響しないはずですが、ディストリビューションとしては大きな変更点です。
- `catalogue-date` 情報は、パッケージアップデートには関係がないことが多いため、TEX カタログには伝達されないことになりました。

9.1.18 2021

全般：

- Donald Knuth による TEX/METAFONT Tuneup 2021 の変更内容を取り込みました（https://tug.org/TUGboat/tb42-1/tb130knuth-tuneup21.pdf）。こうしたコードは `knuth-dist` および `knuth-local` パッケージとして CTAN から入手することもできます。大方の予想通り、この Tuneup による修正は極めて特殊なケースに対するもので、実用的な挙動に関しては何ら影響はありません。
- オリジナルの TEX 以外：`\tracinglostchars` が 3 以上に設定されている場合、存在しない文字が入力された際は単にメッセージをログファイルに出力するだけではなく、エラーを起こすようになりました。また当該文字コードは 16 進数で表示されます。
- オリジナルの TEX 以外：新しい整数パラメタ `\tracingstacklevels` が導入されました。その値が正かつ、`\tracingmacros` の値も正の場合、関連するログ出力でマクロ展開の深さを示すプレフィックスが付くようになります（例えば、深さ 2 のときは `~..`）。また、マクロ展開のトレースはこのパラメタで指定された数値以上に深いものについては切り捨てられます。

ℵ（Aleph）：ℵ 上の LATEX フォーマットである lamed は TEX Live から除かれました。`aleph` バイナリそのものは TEX Live に残り、またサポートも続きます。

LuaTEX:

- Lua インタプリタのバージョンが 5.3.6 になりました。
- 新しい `\tracingstacklevels` の一般化として、`\tracingmacros` で用いられるネストレベルへのコールバックが追加されました。
- 数式グリフはテキストとして処理されないようにマークされました。
- 伝統的な数式コードパスのために width/id 補正をやめました。

METAPOST:

- 再現可能な出力を行うため、環境変数 `SOURCE_DATE_EPOCH` がサポートされました。
- `mpto` で不要な `%` が付いていたバグが修正されました。
- `-T` オプションをドキュメント化したほか、マニュアルに細かい修正を行いました。
- `mp_solve_rising_cubic` が期待通り動作するように、2 進モードと 10 進モードにおける `epsilon` の値を変更しました。

pdfTEX:

- 新プリミティブ `\pdfrunninglinkoff`, `\pdfrunninglinkon` を導入しました。これらは例えばヘッダーやフッターでリンクが生成されるのを防止するのに使うことができます。
- `\pdfendlink` が `\pdfstartlink` と異なるネストレベルで現れた場合に、停止する代わりに警告を発するようになりました。
- `\pdfglyphtounicode` の代入結果を `fmt` ファイルにダンプするようになりました。
- ソース：upstream への追随が困難なので、`poppler` サポートは除去されました。TEX Live では、pdfTEX は常に `libs/xpdf`（調整された `xpdf` の内部ライブラリ）が使われてきました。

XETEX: 数式カーニングに関する修正をしました。

DVIPDFMx:

- Ghostscript はデフォルトで安全に呼び出されるようになりました。(すべての入力ファイルが信頼できるという前提の下で) この安全な呼び出しを無効化する場合は `-i dvipdfmx-unsafe.cfg` を使用してください。これは PSTricks を XeTeX で使用する場合には必須です。すなわち `xetex -output-driver="xdvipdfmx -i dvipdfmx-unsafe.cfg -q -E" ...` としてください。
- 画像ファイルが見つからなかった場合、失敗ステータスで終了します。
- カラーサポートのための special 書式を拡張しました。
- `ExtGState` を操作するための special を追加しました。
- `pdfcolorstack`, `pdffontattr` と互換の special を追加しました。
- `dviluatex` の拡張 `fnt_def` を試験的にサポートしました。
- 仮想フォントに関する新機能として日本語フォント定義のフォールバックを実装しました。

Dvips:

- デフォルトの PostScript 文書のタイトルは入力ファイルのベースネームになりました。またこの設定は新オプション `-title` でも上書きすることができます。
- EPS やその他の画像ファイルが見つからない場合、失敗ステータスで終了します。
- 仮想フォントに関する新機能として日本語フォント定義のフォールバックを実装しました。

MacTeX: MacTeX とその新しいバイナリフォルダ universal-darwin は macOS 10.14 以上 (Mojave, Catalina, Big Sur) を要求するようになりました。x86_64-darwin バイナリフォルダは廃止されました。x86_64-darwinlegacy は Unix 版の `install-tl` でのみ利用可能で、macOS 10.6 以上をサポートしています。

今年は Mac にとっては重要な年です。というのも、Apple が新しい ARM マシンを 11 月に発表し、今後何年も ARM マシンと Intel マシンの販売・サポートを行うとしたからです。universal-darwin にあるすべてのプログラムは ARM と Intel の両方に対応した実行ファイルを持っています。いずれのバイナリも同じソースからコンパイルされています。

MacTeX の追加プログラムである Ghostscript, LaTeXiT, TeX Live Utility, TeXShop はいずれも両方のアーキテクチャに対応しており、また署名済みの hardened runtime を持つようになったので、すべて今年の MacTeX に含められました。

`tlmgr` とインフラ：

- メインリポジトリの texlive.tlpdb バックアップを 1 つに限定しました。
- システム・Perl バージョン依存性をさらに軽減しました。
- tlmgr info が新しいエントリ lcat-* と rcat-*（それぞれローカル・リモートカタログに対応）を表示するようになりました。
- サブコマンドの完全なログは texmf-var/web2c/tlmgr-commands.log に移動しました。

9.1.19 2022

全般：

- 独自の形式 HINT を出力する HiTEX が追加されました。HINT は技術文書をモバイル端末で読むのに適するようにデザインされたものです。HINT ビューアは GNU/Linux, Windows, Android 向けのものが開発されていますが、TEX Live には含まれていません。
- tangle, weave: 3 つ目の引数（省略可）に出力先のファイルを指定できるようになりました。
- WEB プログラムについて「ミニインデックス」を作成するための Knuth 製のプログラム twill が収録されました。

エンジン共通の拡張（Knuthian TEX, Aleph, HiTEX を除く）：

- \show の出力をファイルにリダイレクトする \showstream プリミティブが追加されました。
- 空行や垂直ボックスの終端などに挿入される \par トークンを別のトークンに変更するためのプリミティブ \partokenname, \partokencontext が追加されました。

ε-pTEX, ε-upTEX:

- 新プリミティブ \lastnodefont, \suppresslongerror, \suppressoutererror, \suppressmathparerror が追加されました。
- pdfTEX の拡張 \vadjust pre がサポートされました。

LuaTEX:

- 構造化 PDF 2.0 出力がサポートされました。
- PDF 2.0 の PNG /Smask もサポートされました。
- バリアブルフォントのインターフェースが luahbtex に追加されました。
- \mathdefaultsmode のデフォルトラジカルスタイルが変更されました。
- 選択した discretionary の作成しないようにするオプション機能が追加されました。
- TrueType フォント実装を改善しました。

- \fontdimen アロケーションがより効率的になりました。
- 方向同期ノードが後ろに続くローカル par ノードのみからなる段落は無視するようになりました。

METAPOST: マクロ展開が停止しなくなるバグを修正しました。

pdfTEX:

- 構造化 PDF 2.0 出力がサポートされました。
- レタースペーシング済みフォントについて、指定のある場合は明示的に \fontdimen6 を使用するようになりました。
- 警告は常に行頭に出力されるようになりました。
- 自動カーニング（\pdfappendkern と \pdfprependkern）についても protrusion を行うようになりました。同様に自動カーニングの位置にも明示的もしくは暗黙的なハイフネーションが行われます。

pTEX など：

- 現在の LATEX をより強力にサポートする pTEX 4.0.0 にメジャーアップデートされました。
- 新プリミティブ \ptexlineendmode, \toucs が追加されました。
- これまでは upTEX, ε-upTEX でのみサポートされていた \ucs が pTEX と ε-pTEX でもサポートされました。
- 北川氏の TUGboat 記事（https://tug.org/TUGboat/tb41-3/tb129kitagawa-char.pdf）にあるように、8-bit 文字と和文文字が区別されるようになりました。

XETEX: XETEX と PSTricks の透過機能（Ghostscript の実装の問題で、この機能を使うと安全性が損なわれます）を必要とする文書を簡単に処理できるように、新たなラッパ xetex-unsafe と xelatex-unsafe が追加されました。

DVIPDFMx:

- -dNOSAFER を指定しなくても PSTricks をサポートするようになりました（透過機能を除く）。
- ビットマップフォントの解像度を指定する -r が再び動作するようになりました。

Dvips: デフォルトでは回転されたページサイズ用の自動メディア調整を試行しないようになりました。新設の --landscaperotate オプションを指定すると再度有効化されます。

upmendex: 実験的にアラビア文字・ヘブライ文字をサポートしました。全般的に文字分類と多言語サポートが強化されました。

Kpathsea: kpsewhich -all の結果において通常（非 -all）の探索の結果が一番に出

力されるようになりました。

tlmgr とインフラ：

- mirror.ctan.org についてデフォルトで HTTPS を使うようになりました。
- 再配置が簡単になるように SELFAUTOPARENT の代わりに TEXMFROOT を使うようになりました。
- install-tl: あるパッケージのダウンロードやインストールに失敗した際、自動的に後続のパッケージのインストールを続行し、後で（一度だけ）再試行するようになりました。

MacTEX: MacTEX とそのバイナリフォルダ universal-darwin が macOS 10.14 以上（Mojave, Catalina, Big Sur, Monterey）になりました。x86_64-darwinlegacy フォルダのバイナリ（Unix 版の install-tl でのみインストール可能）macOS 10.6（Snow Leopard）以降をサポートします。

プラットフォーム：サポート対象のプラットフォームに増減はありません。来年（2024 年）のリリースからは Windows 向けのバイナリを 32-bit から 64-bit に変更する予定です。残念ながら両方を同時にうまくサポートすることはできません。

9.1.20 2023

Windows: 以前からアナウンスしていた通り、TEX Live は 32-bit バイナリの代わりに 64-bit バイナリを収録するようになりました。新しいディレクトリ名は bin/windows です（64-bit バイナリを名前に “32” と付くディレクトリに配置するのは不適当に思われたため）。この変更が Windows ユーザに余計な仕事を発生させることは承知の上ですが、これよりも良い方法は見つかりませんでした。32-bit Windows のサポートが必要な方は、TEX Live2022 をインストールする必要がある場合があります。Windows 11 以降では、ARM 上の Windows は 64-bit AMD/Intel をエミューレートするので最新の TEX Live を利用することができます。一方、ARM 上の Windows 10 は 64-bit AMD/Intel をエミューレートできないため、32-bit TEX Live が必要です。詳細は TEX Live の Windows 向け特別ページ（https://tug.org/texlive/windows.html）も参照してください。

TEX エンジン共通の拡張（オリジナルの TEX と ε-TEX を除く）：\special の直後に新しいキーワード “shipout” が置かれた場合、その引数トークンの展開が \shipout 時（すなわち \immediate なしの \write と同じタイミング）まで遅延されるようになりました。

ε-pTEX と ε-upTEX:

- 素の (u)pTEX はビルドされなくなりました。今後 ptex, uptex コマンドはそれぞ

れ eptex, euptex の互換モードを起動します。pTEX tools についても同様です（後述）。

- 新プリミティブ：\tojis, \ptextracingfonts, \ptexfontname。
- \font で新しい JIS/UCS シンタックスがサポートされました。

LuaTEX:

- ファミリ適用を許容したまま、数式文字のクラスを保持するための新プリミティブ \variablefam。
- r2l アノテーション領域の改善。
- エンジン共通の遅延 \special（上述）。
- LuaTEX 1.17.0 が 5 月にリリースされました：socket ライブラリはデフォルトでは無効化されました。新しいコマンドラインオプション --socket または --shell-escape（--shell-restricted ではありません）を利用すると socket が有効になります。また mime ライブラリは常に利用可能になりました。新しい関数 os.socketsleep と os.socketgettime も常に利用可能です。その他、セキュリティの向上も図られました。

METAPOST: バグ修正を行いました。精度改善のため svg->dx と svg->dy は double になりました。mp_begin_iteration が更新されました。mplib のメモリリークが修正されました。

pdfTEX:

- /Info 辞書を完全に省略するための新プリミティブ \pdfomitinfodict。
- /ProcSet 配列の省略をコントロールする新プリミティブ \pdfomitprocset：このパラメタの値が負のとき、または値がゼロで pdfTEX が PDF 1.x 出力を生成中の場合は /ProcSet が含められます。
- \pdfinterwordspaceon 下において、現在のフォントエンコーディングが /space 文字をスロット 32 に持っている場合はそれが利用されます。そうでない場合は /space には（新規の）デフォルトフォント pdftexspace が用いられます。このデフォルトフォントは新プリミティブ \pdfspacefont によって上書きすることが可能です。この新手順は \pdffakespace でも用いられます。

pTEX など：

- 前述の通り、ptex は独自のバイナリではなく eptex の互換モードを起動するようになりました。
- pTEX tools（pbibtex, pdvitype, ppltotf, ptftopl）はそれぞれ upTEX バージョンの対応ツールに統合され、互換モードで動作するようになりました。

XeTeX: \XeTeXupwardsmode が有効になっている際の \topskip と \splittopskip の計算に関するバグ修正と、上述のエンジン共通の遅延 \special 追加が行われました。

DVIPDFM*x*: pdfmark/bookmark を作成する新オプション --pdfm-str-utf8 が追加されました。

BibTeXu:

- この BibTeX 拡張は、オリジナルよりも遥かに強力な（Unicode ベースの）多言語サポートを備えた BibTeX のほぼ完全上位互換プログラムです。TeX Live には数年前に取り込まれました。
- 今年は CJK 言語をサポートするための機能がさらに追加されました。また日本語組版用の (u)pbibitex その他のプログラムと同様の拡張も行われました。

Kpathsea: Windows 版と同様、Unix 系プラットフォームでもファイルエンコーディング推定をサポートしました。(e)p(la)tex と pbibtex で有効になっています。

tlmgr とインフラ：

- macOS でのデフォルトをテキストモードに変更しました。
- コアパッケージを先にインストールし、それ以外のパッケージのリトライは 1 回になりました。
- ディスクの空き容量が十分かどうか、簡易チェックを行うようになりました。

MacTeX:

- MacTeX とそのバイナリディレクトリ universal-darwin の要求は macOS 10.14 以上になりました（Mojave, Catalina, Big Sur, Monterey, Ventura）。Unix 版の install-tl でのみ利用可能な x86_64-darwinlegacy バイナリディレクトリは macOS 10.6（Snow Leopard）以降をサポートしています。
- MacTeX の GUI パッケージに、macOS 向けの HINT 文書[*14]ビューア hintview が追加されました。GUI パッケージはドキュメントフォルダはインストールしなくなりました。代わりに、新規ユーザのための短い READ ME と hintview に関するページが追加されました。
- DVD の Extra ディレクトリ内の追加 TeX ソフトウェアは、ダウンロードサイトのリンクを含むドキュメントに置き換えられました。

プラットフォーム：

- 前述の通り、新しい windows バイナリディレクトリは 64-bit Windows バイナリ

[*14] hitex や hilatex エンジンで作成される、モバイル向け文書。詳細は HiTeX のウェブサイト（https://hint.userweb.mwn.de/hint/hitex.html）を参照してください。

を格納しています。

- bin/win32 バイナリディレクトリは削除されました。32-bit と 64-bit の両方を同時にサポートすることはできないためです。
- i386-cygwin バイナリディレクトリも削除されました。Cygwin が i386 のサポートを終了したためです。

9.2 現在：2024

エンジン共通の拡張（Knuthian TEX, ε-TEX, LuaTEX を除く）：

- \output と \write のブレースの不正利用によるクラッシュを防止。
- 内部ループのわずかな最適化（いずれの変更も David Fuchs 由来）。

LuaTEX:

- ファイル書き込み時の Lua レベルチェックを TEX レベルチェックと同等に。
- Lua 変数 texconfig.trace_extra_newline が設定されている場合、ログファイルの各マクロパラメタの間に改行を出力。2020–2023 年はこれが規定の挙動でした。それ以前は、他のエンジンと同様このような空行は一切出力されませんでした。

METAPOST: 区間演算のための MPFI ライブラリが導入されました。このライブラリは --numbersystem=interval で有効化されます。

pdfTEX:

- ビッグエンディアンシステムにおけるテキスト抽出の信頼性向上。
- 破損した PFB, FMT, PDF ファイルに対するチェックの強化。
- 異常な PDF のインクルードがあった場合に、破損した PDF を生成するのを抑止。
- マニュアルを LATEX に変換し、校正も実施。ただし重要な更新はわずかです。

ε-pTEX, ε-upTEX:

- 新しいシンプルなエンコーディング変換ユーティリティ ptekf。
- eptex の独立バイナリを廃止。このコマンドは、今後は euptex のレガシーエンコーディングモードを起動するようになります。

XETEX: BMP 範囲外の Unicode 文字を含むファイル名をサポート。

DVIPDFMx:

- pdfTEX と同じ方法でタイムゾーンオフセットを計算。
- 新しい Adobe-Japan1-UCS2 CMap に対応するために異体字セレクタを処理。
- (u)pTEX のため仮想フォントのマルチバイト文字を OFM フォントにフォール

バック。

Kpathsea:

- Lua(LaTeX に向けて、TEXMF(SYS)VAR 以下への書き出しも許可するべくファイル名の拡張された安全性チェックをサポート。これにともない、新しい関数と対応する kpsewhich のオプションを追加。
- 新しい環境変数 TEXMF_OUTPUT_DIRECTORY を追加（従来の TEXMFOUTPUT とは別）。これによりサブプログラムが --output-directory 設定にアクセスできるようになります。

CWEB:

- バックエンド TeX エンジンとして LuaTeX をサポート。
- ブレークポイントのない行におけるバッファオーバーフローを修正。

tlmgr:

- バグ報告情報にアクセスするための新アクション tlmgr bug。
- LWP サポートを強化するため（特に macOS で重要）、独自のルート証明書バンドルをすべてのプラットフォームで利用。
- ローカルの TeX Live より古いリモートリポジトリ（つまりミラーの状態）を検出した場合はアップデートを行わないことで、誤ったパッケージ削除を防止。
- ConTEXt の紙面サイズ設定をサポート。
- ミラーリポジトリ（main と tlcontrib）用の tlpdbsake バックアップを除去。

MacTeX: 最新版の Ghostscript 10.02.1 を同梱。4 つの GUI プログラムと最新 TeX Live を収録。バイナリは macOS 10.14（Mojave, 2018）以降のサポートを維持（いつも通り、古い Mac は TeX Live に含まれる x86_64-darwinlegacy バイナリではサポートされていますが、MacTeX ではサポート外です）。

9.3 未来

私たちはこれからも新しいバージョンのリリースを続けるつもりで、より多くのドキュメントとプログラム、これまで以上に良質で品質の保証されたマクロとフォントのツリー、その他 TeX に関連するあらゆる成果物を提供していくつもりです。こうした仕事はすべてボランティアが余暇を使用して行っており、そのために常にやるべきことが残っています。ぜひ https://tug.org/texlive/contribute.html を読んで、コントリビューションを検討してください。

間違いの指摘や提案、ヘルプの要望は下記にご連絡ください。メーリングリストを購読せずに投稿しても構いませんが、英語で書いてください。

tex-live@tug.org
https://tug.org/texlive

Happy TEXing!