

1. OSS の概要に関する知識 I

1. 科目の概要

オープンソースソフトウェアの基本的な概念や歴史、代表的な利用方法を解説し、効果的な活用方法と OSS プロジェクトへの参加方法、コミュニティとの関係について説明する。

2. 習得ポイント

本科目の学習により習得することが期待されるポイントは以下の通り。

習得ポイント	説明	シラバスの対応コマ
I-1-1. オープンソースソフトウェア(OSS)とは？	オープンソースソフトウェア(OSS)という言葉の定義とOSSの概念を説明する。また、なぜOSSが注目を浴びているのか、OSSが登場した背景や理念について説明し、さらにOSSの開発モデルやライセンスといったトピックについて紹介する。	1
I-1-2. UNIXからLinuxへの歴史	1960年代後半に始まるUNIXの歴史から、BSDとしての発展、Linuxの登場と展開といった流れを紹介する。また関連する話題としてGNUプロジェクトやFSF(フリーソフトウェア財団)について説明する。	2
I-1-3. 各種のOSSサーバ	OSSが最も広く利用されている分野として、インターネットサーバ(メール、Web、DNS)、各種ネットワークサーバ(ファイル共有、ログ、認証)があり、OSSのミドルウェア(データベースサーバ、アプリケーションサーバ)も普及しつつあることを説明する。最近注目されつつある。OSSの仮想化ツールについても簡単に紹介する。	3
I-1-4. OSSの開発ツール	OSSを利用して開発する、あるいはOSSを開発するための、開発言語として、C/C++、Java、PHP、Perl、Python、Rubyを紹介する。また、PHP、Ruby、Javaそれぞれに対するOSS開発フレームワークを紹介し、OSS統合開発ツールとしてEclipse、NetBeans、WideStudioの概要を紹介する。	4,5
I-1-5. OSSのデスクトップアプリケーション	OSSによるデスクトップアプリケーションの例として、代表的なウィンドウシステム(統合デスクトップ環境)、オフィススイート、ブラウザ、メール、グラフィックツールを紹介する。	5
I-1-6. OSSのサーバアプリケーション	OSSによるサーバアプリケーションとして、最近注目を集めているCRM(顧客管理)、BI(企業内統合データ分析)、ERP(経営資源管理)、CMS(Webサイトのコンテンツ管理)向けのOSSを紹介する。また特定業務向けのOSSアプリケーションの例も紹介する。	5
I-1-7. OSSの標準化動向と標準化の意義	OSSの標準化動向を概観する。Linuxの標準規格であるLSB(Linux Standard Base)や、Javaの標準仕様を紹介し、標準を定める意義を解説する。またアジア地域での標準化への取り組みや国際標準機関との関係についても触れる。	6
I-1-8. OSSの利用状況	ビジネスにおけるOSSの利用状況を各種調査結果を参照して概説する。LinuxおよびOSSミドルウェアの市場シェア、企業ユーザのOSS導入意向やメリット・デメリットを感じる点に触れる。	6,7
I-1-9. OSSによるWebシステムの構築	典型的なWebシステムの構築を例に、OSSの使われ方、実際のシステム構築上のポイントやメリット・デメリット、注意点などを説明する。	8
I-1-10. OSSコミュニティの種類と特徴、参加方法	OSSコミュニティの代表的な事例を紹介し、OSSコミュニティの種類と特徴を概観する。またOSSコミュニティへの参加方法や参加時の留意点について説明する。	8

【学習ガイダンスの使い方】

- 「習得ポイント」により、当該科目で習得することが期待される概念・知識の全体像を把握する。
- 「シラバス」、「IT 知識体系との対応関係」、「OSS モデルカリキュラム固有知識」をもとに、必要に応じて、従来の IT 教育プログラム等との相違を把握した上で、具体的な講義計画を考案する。
- 習得ポイント毎の「学習の要点」と「解説」を参考にして、講義で使用する教材等を準備する。

3. IT 知識体系との対応関係

「1. OSS の概要に関する知識 I」と IT 知識体系との対応関係は以下の通り。

科目名	基本レベル (I)								応用レベル (II)						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1. OSS の概要に関する知識	<オープンソースの理念>	<オープンソースOSの歴史>	<代表的なオープンソース>	<代表的なオープンソース開発言語>	<代表的なオープンソースアプリケーション>	<オープンソースの市場動向>	<オープンソースソフトウェアを用いたシステム事例>	<オープンソースソフトウェアコミュニティ>	<オープンソースソフトウェアビジネス>	<オープンソースの技術情報獲得方法>	<オープンソースの技術導入と動作確認>	<オープンソースのOSSの導入と動作確認>	<オープンソースのデスクトップ用アプリケーションの導入と動作確認>	<オープンソースのサーバサイドアプリケーションの導入と動作確認>	<オープンソースの仮想化ツールの導入と動作確認>

[シラバス : http://www.ipa.go.jp/software/open/ossce/download/Model_Curriculum_05_01.pdf]

<IT 知識体系上の関連部分>

分野	科目名	1	2	3	4	5	6	7	8	9	10	11	12	13	
情報処理系標準と情報セキュリティ	1	IT-IAS 情報保護と情報セキュリティ	IT-IAS1. 基礎的問題	IT-IAS2. 情報セキュリティの仕組み(対策)	IT-IAS3. 運用上の問題	IT-IAS4. ホリゾ	IT-IAS5. 攻撃	IT-IAS6. 情報セキュリティ分析	IT-IAS7. フォレンジック(情報保護)	IT-IAS8. 情報の保護	IT-IAS9. 情報セキュリティサービ	IT-IAS10. 脅威分析モデル	IT-IAS11. 脆弱性		
	2	IT-SP 社会的な観点とグローバルな視点としての課題	IT-SP1. プロフェッショナルとしてのコミュニケーション	IT-SP2. コンピュータの歴史	IT-SP3. コンピュータを取り巻く社会環境	IT-SP4. チームワーク	IT-SP5. 知的財産権	IT-SP6. コンピュータの法的問題	IT-SP7. 組織のIT	IT-SP8. プロフェッショナルとしての倫理的な問題と責任	IT-SP9. プライバシーと個人の自由				
応用技術	3	IT-IM 情報管理	IT-IM2. データベース関係の概念と基礎	IT-IM3. データアーキテクチャ	IT-IM4. データモデリングとデータベース設計	IT-IM5. データと情報の管理	IT-IM6. データベースの応用分野								
	4	IT-WS Webシステムとその技術	IT-WS1. Web技術	IT-WS2. 情報アーキテクチャ	IT-WS3. デジタルメディア	IT-WS4. Web開発	IT-WS5. 脆弱性	IT-WS6. ソーシャルソフトウェア							
ソフトウェアの方法と技術	5	IT-PF プログラミング基礎	IT-PF1. 基本プログラミングの要素	IT-PF2. プログラミングの基本的構成要素	IT-PF3. オブジェクト指向プログラミング	IT-PF4. アルゴリズムと問題解決	IT-PF5. イベント駆動プログラミング	IT-PF6. 再帰							
	6	IT-PT 技術を統合するためのプログラミング	IT-PT1. システム間連携	IT-PT2. データ取り扱いと交換	IT-PT3. 統合的コーディング	IT-PT4. スクリプティング手法	IT-PT5. ソフトウェアセキュリティの要約	IT-PT6. 種々のプログラミング言語の概要	IT-PT7. ログ						
	7	DE-SME ソフトウェア工学	DE-SME0. 歴史と概要	DE-SME1. ソフトウェアプロセス	DE-SME2. ソフトウェアの要求と仕様	DE-SME3. ソフトウェアの設計	DE-SME4. ソフトウェアのテストと検証	DE-SME5. ソフトウェアの保守	DE-SME6. ソフトウェア開発・検証と連携	DE-SME7. ソフトウェアプロジェクト管理	DE-SME8. 言語圏	DE-SME9. ソフトウェアのフォーマルトトランス	DE-SME10. ソフトウェアの構成管理	DE-SME11. ソフトウェアの標準化	
	8	IT-SIA システムインテグレーションとアーキテクチャ	IT-SIA1. 要求仕様	IT-SIA2. 調達/手配	IT-SIA3. インテグレーション	IT-SIA4. プロジェクト管理	IT-SIA5. テストと品質保証	IT-SIA6. 組織の特性	IT-SIA7. アーキテクチャ						
システム基盤	9	IT-NET ネットワーク	IT-NET1. ネットワークの基礎	IT-NET2. ルーティングとスイッチング	IT-NET3. 物理層	IT-NET4. セキュリティ	IT-NET5. アプリケーション分野	IT-NET6. ネットワーク管理							
	10	DE-NWK テレコムネットワークシステム	DE-NWK0. 歴史と概要	DE-NWK1. 通信ネットワークのアーキテクチャ	DE-NWK2. 通信ネットワークのプロトコル	DE-NWK3. LANとWAN	DE-NWK4. クラウドサービス/パブリッククラウド	DE-NWK5. データのセキュリティと整合性	DE-NWK6. ワイヤレス通信・モバイル通信とモバイルコンピューティング	DE-NWK7. データ通信	DE-NWK8. 組み込み機器向けネットワーク	DE-NWK9. 通信技術とネットワーク概要	DE-NWK10. 性能評価	DE-NWK11. ネットワーク管理	DE-NWK12. 圧縮と伸張
	11	IT-PI プラットフォーム技術	IT-PI1. オペレーティングシステム	IT-PI2. アーキテクチャと連携	IT-PI3. コンピューティングプラットフォーム	IT-PI4. デバイスマネジメントソフトウェア	IT-PI5. ファームウェア	IT-PI6. ハードウェア							
クラウドコンピューティング	12	DE-OPS オペレーティングシステム	DE-OPS0. 歴史と概要	DE-OPS1. 並行性	DE-OPS2. スケジューリングとアーキテクチャ	DE-OPS3. メモリ管理	DE-OPS4. セキュリティと保護	DE-OPS5. ファイル管理	DE-OPS6. リアルタイムOS	DE-OPS7. OSの概要	DE-OPS8. 設計の原則	DE-OPS9. デバイスマネジメント	DE-OPS10. システム性能評価		
	13	DE-CAD コンピュータアーキテクチャと構成	DE-CAD0. 歴史と概要	DE-CAD1. コンピュータアーキテクチャの基礎	DE-CAD2. メモリシステムの構成とアーキテクチャ	DE-CAD3. インタフェースと通信	DE-CAD4. デバイスサブシステム	DE-CAD5. CPUアーキテクチャ	DE-CAD6. 性能・コスト評価	DE-CAD7. 分散・並列処理	DE-CAD8. コンピュータによる計算	DE-CAD9. 性能向上	DE-CAD10. ネットワーク		
複数領域にまたがるもの	14	IT-ITF IT基礎	IT-ITF1. ITの一般的なテーマ	IT-ITF2. 組織の問題	IT-ITF3. ITの歴史	IT-ITF4. IT分野(学)とそれに関連する分野(学)	IT-ITF5. 応用領域	IT-ITF6. IT分野における数学と統計学の活用							
	15	DE-ESY 組み込みシステム	DE-ESY0. 歴史と概要	DE-ESY1. 低電力コンピュータ設計	DE-ESY2. 高信頼性システムの設計	DE-ESY3. 組み込み用アーキテクチャ	DE-ESY4. 開発環境	DE-ESY5. ライフサイクル	DE-ESY6. 要件分析	DE-ESY7. 仕様定義	DE-ESY8. 構造設計	DE-ESY9. テスト	DE-ESY10. プロジェクト管理	DE-ESY11. 並行設計(ハードウェア、ソフトウェア)	DE-ESY12. 実装

4. OSS モデルカリキュラム固有の知識

OSS モデルカリキュラム固有の知識として、OSS の開発モデルとライセンス、Unix と Linux、OSS の利用事例、OSS の開発プロジェクトとコミュニティなどに関するものがある。また、具体的な OSS の統合開発ツール（開発環境）についての知識を習得する。

科目名	第1回	第2回	第3回	第4回	第5回	第6回	第7回	第8回
1. OSS の概要に関する知識 I	(1) オープンソースの登場と理念 (2)「伽藍とバザール」モデル (3) オープンソースのライセンス	(1) UNIX の誕生と発展 (2) GNU・コピーレフト (3) Linux	(1) OS(オペレーティングシステム) (2) インターネットサーバ (3) 各種サーバ (4) ミドルウェア (5) 仮想化ツール	(1) 開発言語 (2) 開発フレームワーク (3) 統合開発ツール	(1) デスクトップアプリケーション (2) サーバアプリケーション	(1) 標準化動向 (2) 利用分野/導入事例	(1) Web システムのアーキテクチャ適用事例 (2) 構築上のポイント (3) システムの効果と問題点	(1) プロジェクトの発達と運営 (2) オープンソースソフトウェアコミュニティへの参加 (3) 代表的なコミュニティ

(網掛け部分は IT 知識体系で学習できる知識を示し、それ以外は OSS モデルカリキュラム固有の知識を示している)

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-1. オープンソースソフトウェア(OSS)とは？	
対応する コースウェア	第 1 回 (オープンソースの理念)	

I-1-1. オープンソースソフトウェア(OSS)とは？

オープンソースソフトウェア(OSS)という言葉の定義と OSS の概念を説明する。また、なぜ OSS が注目を浴びているのか、OSS が登場した背景や理念について説明し、さらに OSS の開発モデルやライセンスといったトピックについて紹介する。

【学習の要点】

- * オープンソースソフトウェアはソースコードが公開されているため、誰でも自由に利用可能で、誰でもソースコードの解析・研究が可能である。
- * OSS は、普及の中で標準化が進んでいることや、投資コストの低価格化などに期待が集まって、近年注目を浴びている。
- * コミュニティが開発を進めている OSS や、企業が開発を終えたあとに OSS としてコミュニティに譲渡するものなどがあり、OSS の対象範囲は拡大しつつある。
- * OSS はソフトウェア開発におけるビジネスモデルの変革を伴う。

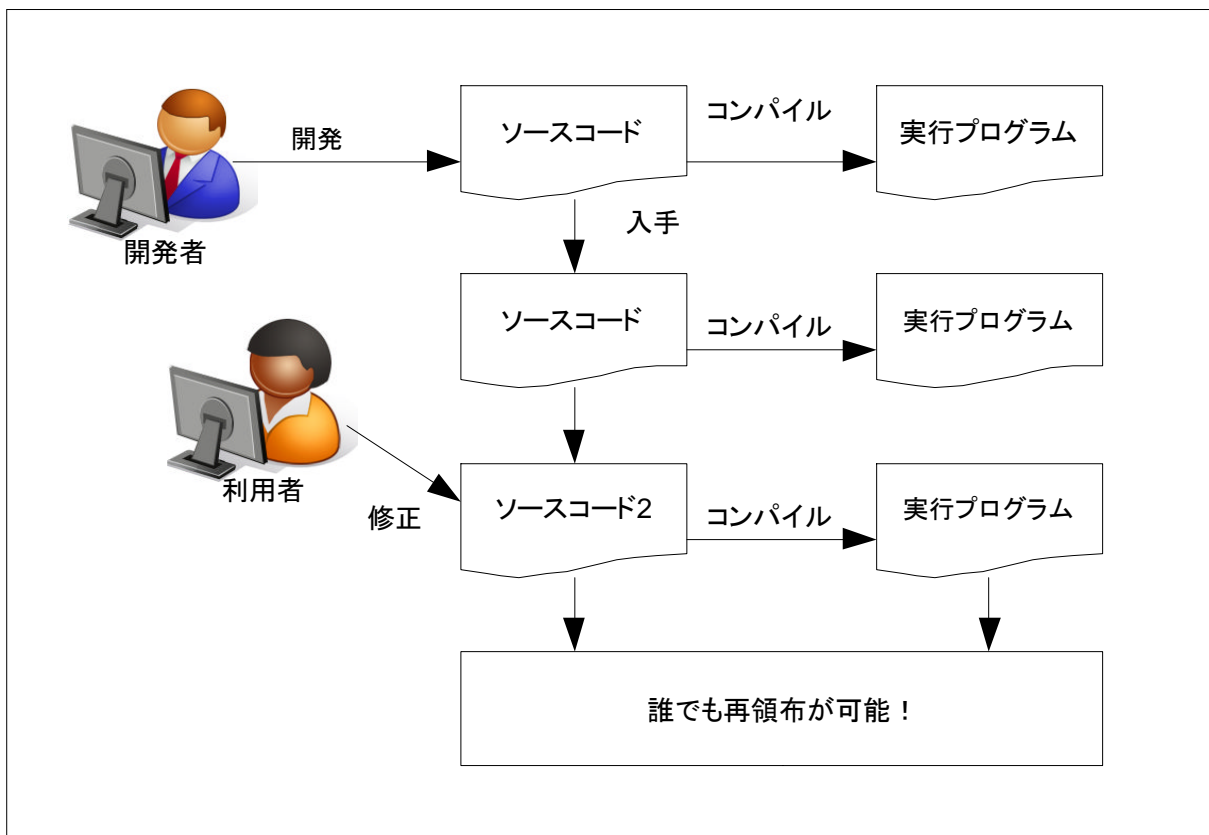


図 I-1-1. OSS の概念

【解説】

1) オープンソースソフトウェアとは何か

オープンソースソフトウェア(以下 OSS)とは、「ソースコードがオープンになっているソフトウェア」のことを指すが、より正確には、OSI(Open Source Initiative)の「The Open Source Definition」によって定義づけられている。OSS のライセンス条件に関しては、以下に示すようなものが代表的なものとなる。

- * ソースコードが入手可能である
 - OSS は、今まで開発元のノウハウとして非公開であったソースコードが公開されている。
 - 誰もが自由にそのソースコードを解読し、研究することができる。
 - 利用者は無償でそれらのソフトウェアを利用することができる。
- * 自由な再頒布が可能である
 - OSS は自由に再頒布することが可能である。
- * 派生物が作成でき、派生物にも同じライセンスを適用することが可能である
 - OSS は、ソースコードを解読して研究することで、派生物を作成することができる。
 - 派生物を他の利用者に再頒布することが可能である。
 - 再頒布した派生物にも、同じライセンスが適用されることになる。
 - さらに他の利用者に再頒布することも可能である。

2) なぜオープンソースソフトウェアが注目を浴びているのか

OSS は近年になって注目され普及し始めたのではない。Apache、bind、gcc などのプログラムは、以前から利用されていた。最近になって急激に OSS が普及してきたようなイメージがあるのは、Linux の認知と普及によるものといえる。

- * コミュニティをベースとして開発が進められている OSS
 - Linux や PostgreSQL などは、コミュニティ主導で開発が行われている。
- * 企業が作成したソースコードが、OSS として提供されているもの
 - MySQL や OpenOffice.org などは、企業が開発し OSS のコミュニティに譲渡された。
- * このような OSS が登場することで、普及の中での標準化が進み、投資コストの低価格化にも拍車がかかり、さらなる OSS の普及が促進されてきた。

3) オープンソースソフトウェアでのビジネスモデル

OSS でビジネスを成功させようとしている企業や、すでに収益をあげている企業も存在するが、まだまだ多くの企業が手がけているとはいえない。

- * SI事業者
 - 公開されているソースコードにより、問題の分析・解決を行うことが可能となる。
 - サポート分野において、他社との差別化が可能となる。
- * ハードウェアベンダ
 - OSS 開発に参加することで、自社で単独開発を行うよりも大きなコスト削減となる。
 - OSS の開発に積極的に協力・利用することで、高い生産性を得られる。

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-2. UNIX から Linux への歴史	
対応する コースウェア	第2回 (オープンソース OS の歴史)	

I-1-2. UNIX から Linux への歴史

1960 年代後半に始まる UNIX の歴史から、BSD としての発展、Linux の登場と展開といった流れを紹介する。また関連する話題として GNU プロジェクトや FSF(Free Software Foundation)について説明する。

【学習の要点】

- * UNIX はオープンな外部仕様をもつ OS として発展してきたが、実装の統一化が不十分なため、様々な亜種を発生させた。
- * ヘルシンキ大学の学生であったリーナス・トーバルズ氏は UNIX の外部仕様に準拠した Linux を一から作成した。
- * Linux は世界中の開発者やベンダーの協力で、現在の主流 OS の一つとして発展してきた。

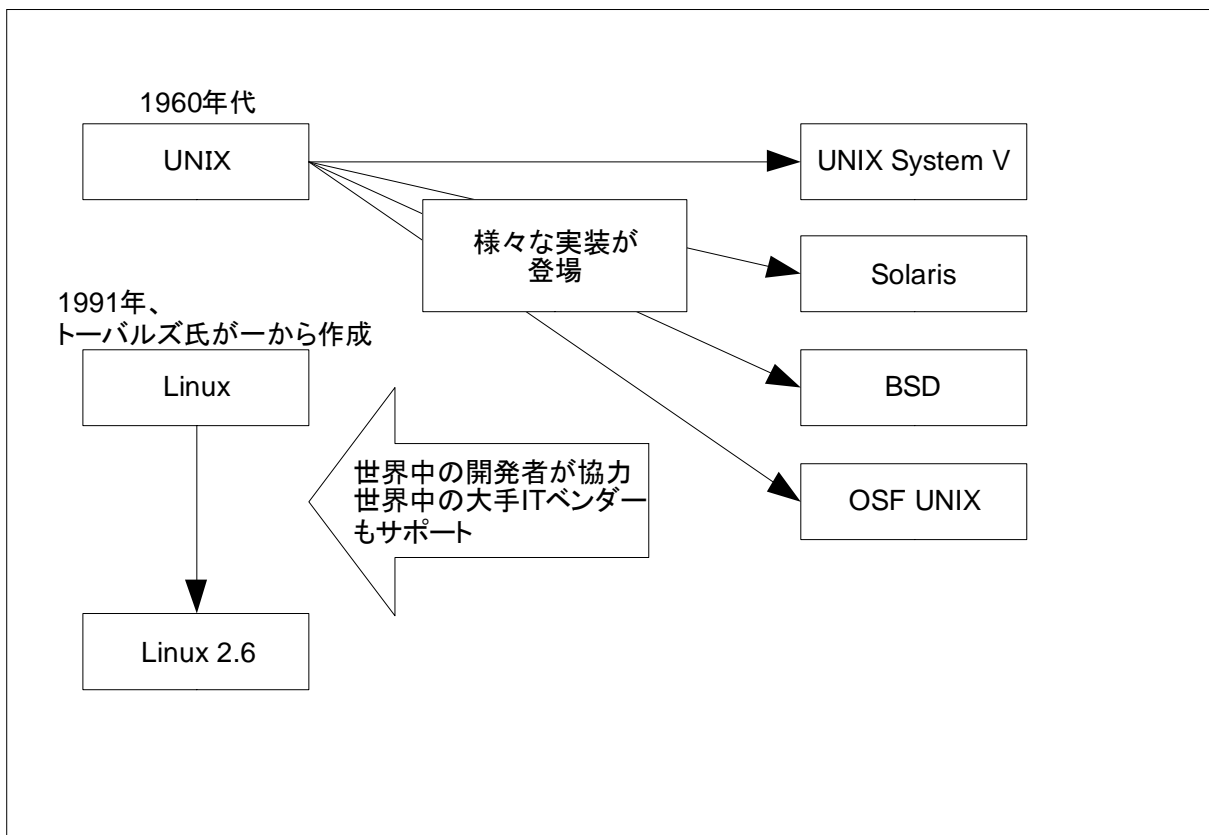


図 I-1-2. UNIX の歴史

【解説】

1) UNIX の歴史

UNIX は 1960 年代に開発され、1967 年には教育機関向けにソースコードも含めて無償公開された。その後、UNIX は様々な亜種を生み出してゆく。

- * 1963 年頃から、TSS(Time Sharing System) を利用した Multics の開発がはじまる。それまでの、効率の悪い処理方法を改善するための処理方法として、CTSS(後の TSS)といった処理方法が開発された。この TSS を利用した Multics の開発が開始された(ただし、Multics は巨大なシステムとなりすぎて成功しなかった)。
- * UNIX のファーストバージョンの誕生
移植のために Multics からは、マルチタスク機能を含む多くの機能が削ぎ落とされたが、あまりにもシンプルすぎることから、マルチユーザなどの機能が追加で盛り込まれて、UNIX のファーストバージョンとなった。
 - アセンブラ言語では移植性に限界があったため、C言語に書き換えられた。
 - 1967 年、UNIX は教育機関向けにソースコードを含め、無償公開された。
- * 1977 年に BSD が誕生する
カリフォルニア大学バークレー校の大学院生が、UNIX 上に実装されていた Pascal コンパイラの改良を開始する。この Pascal コンパイラを利用したいという要望が多く、これに対応するためバークレー版 UNIX ソフトウェア群をまとめたが、これが BSD となった。
- * 1979 年、UNIX 32/V の発表
32 ビット化された UNIX 32/V は、16 ビット版である UNIX V8(以後 V9, V10, Plan9)を機能統合させ、System IIIとして発表される。さらに機能の強化が行われたものが System Vとなる。

2) Linux の登場

1991 年、教育用 UNIX のクローンである Minix のニュースグループに、リーナス・トーバルズ氏が開発した自分専用のオペレーティングシステムのソースコードが公開される。このコードはまだ不完全な状態で公開され、多くの人にデバッグや機能改善を手伝ってもらおうという方法を用いた。一方、GNU プロジェクトとしては、OS の中核となるカーネル部分の開発は空白のままであった。そのタイミングでトーバルズ氏の投稿があり、以後 GNU プロジェクトでは Linux を中核に開発が進められていく。

3) GNU プロジェクトと FSF(Free Software Foundation)

AT&T 社が UNIX の商品化に向けて動き出したことを見て、多くの企業がソフトウェアを独占的なものへと変えてゆく。この状況に反発したユーザ達が GNU プロジェクトを設立する。

- GNU プロジェクトは、フリーソフトウェアでシステム全体を構築することを目的としている。
- GNU プロジェクトとして第一の仕事は、コンパイラの製作であった。
(ただし、コンパイラの作成は時間が掛かるため、Gnu Emacs の作成が優先された)
- Gnu Emacs をコピーしたテープを、150 ドルで発送するというビジネスが、GNU プロジェクトを起動に乗せる。また、GNU ソフトウェアの開発・配布・マニュアル販売などを行うフリーソフトウェア財団(FSF: Free Software Foundation)が設立される。

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-3. 各種の OSS サーバ	
対応する コースウェア	第 3 回 (代表的なオープンソース)	

I-1-3. 各種の OSS サーバ

OSS が最も広く利用されている分野として、インターネットサーバ(メール、Web、DNS)、各種ネットワークサーバ(ファイル共有、ログ、認証)があり、OSS のミドルウェア(データベースサーバ、アプリケーションサーバ)も普及しつつあることを説明する。最近注目されつつある、OSS の仮想化ツールについても簡単に紹介する。

【学習の要点】

- * サーバは、メールサーバ、Web サーバ、DNS サーバ等のインターネットサーバ、ファイルサーバ、ログサーバ、認証サーバ等のネットワーク関連のサーバ、DB サーバ、AP サーバ等のミドルウェアに分類される
- * サーバの役割は様々な種類に分類できるが、そのほとんどの種類で OSS の実装がなされている。
- * OSS におけるサーバの実装としては、メールサーバの sendmail、Web サーバの Apache HTTP Server などが有名である。
- * 1 台のマシン上に複数の仮想マシンを構築し、その各々の仮想マシン上にサーバを設置する、仮想化という技術が近年注目を浴びている。

種類	説明	代表的なOSS	
インターネット サーバ	メールサーバ	メールの送受信を行うサーバ	Sendmail
	Webサーバ	Webページ(ホームページ)を送信するサーバ	Apache
	DNSサーバ	ホスト名からIPアドレスを検索したり、名前解決を担当するサーバ	bind
ネットワーク 関連の サーバ	ファイルサーバ	ネットワーク内のユーザに対して、ファイル共有サービスを提供するサーバ	Samba、NFS
	ログサーバ	ログを収集するサーバ	Syslogd
	認証サーバ	認証サービスを提供するサーバ	NIS
ミドルウェア	DBサーバ	データベースへのアクセスを提供するサーバ	PostgreSQL
	Apサーバ	各種の処理を提供するアプリケーションサーバ	Tomcat、JBoss

図 I-1-3. サーバの種類

【解説】

1) サーバの役割

サーバとは、(クライアントに対して)何らかのサービスを提供する機能あるいはシステムである。サーバはその役割から次のように分類できる。

- * インターネットサーバ
インターネット上のサービスをユーザに提供するサーバ。電子メールの送受信を行うメールサーバ、Web ページを提供する Web サーバ、ホスト名から IP アドレスへの名前解決を行う DNS サーバなど。
- * ネットワーク関連のサーバ
ネットワーク(LAN)内のサービスをユーザに提供するサーバ。ファイル共有を行うファイルサーバ、ログ収集を行うログサーバ、認証情報を提供する認証サーバなど。
- * ミドルウェア
上記サーバと OS(オペレーティングシステム)との中間に位置づけられるサーバ。DB(データベース)へのアクセスを提供する DB サーバ、Web サーバ等からの要求を処理する AP(アプリケーション)サーバなど。

2) OSS におけるサーバの実装

OSS の代表的なサーバは以下の通りである。

- * メールサーバ: sendmail、Postfix、Dovecot、Courier-IMAP
- * Web サーバ: Apache HTTP Server
- * DNS サーバ: BIND
- * ファイルサーバ: Samba、NFS
- * ログサーバ: syslog、syslog-ng
- * 認証サーバ: NIS
- * DB サーバ: MySQL、PostgreSQL、Firebird
- * AP サーバ: Tomcat、JBoss
- * その他のサーバ: Squid、vsftpd

3) 仮想化とは

1 台のマシンで動作させるソフトウェア上で、複数の仮想マシンを構築する技術であり、仮想化を実現するソフトウェアを仮想マシンモニタと呼ぶ。リソース(CPU、メモリ、ディスク等)は仮想マシンモニタにより各仮想マシンで共有されるか、仮想マシンへ割り当てられる。OSS の代表的な仮想マシンモニタ製品として、Xen が挙げられる。

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-4. OSS の開発ツール	
対応する コースウェア	第4回（代表的なオープンソース開発言語） 第5回（代表的なオープンソースアプリケーション）	

I-1-4. OSS の開発ツール

OSS を利用して開発する、あるいは OSS を開発するための、開発言語として、C/C++、Java、PHP、Perl、Python、Ruby を紹介する。また、PHP、Ruby、Java それぞれに対する OSS 開発フレームワークを紹介し、OSS 統合開発ツールとして、Eclipse、NetBeans、WideStudio の概要を紹介する。

【学習の要点】

- * 開発言語には様々なものがあり、オープンソースソフトウェアとして提供されているものもある。
- * 繰り返し必要とされる機能をまとめて提供し、アプリケーションソフトウェアのベースを作成するソフトウェアをフレームワークと呼ぶ。
- * 高度なエディタ機能やデバッグ支援機能、バージョン管理機能などを実現する統合開発環境もオープンソースソフトウェアとして提供されている。

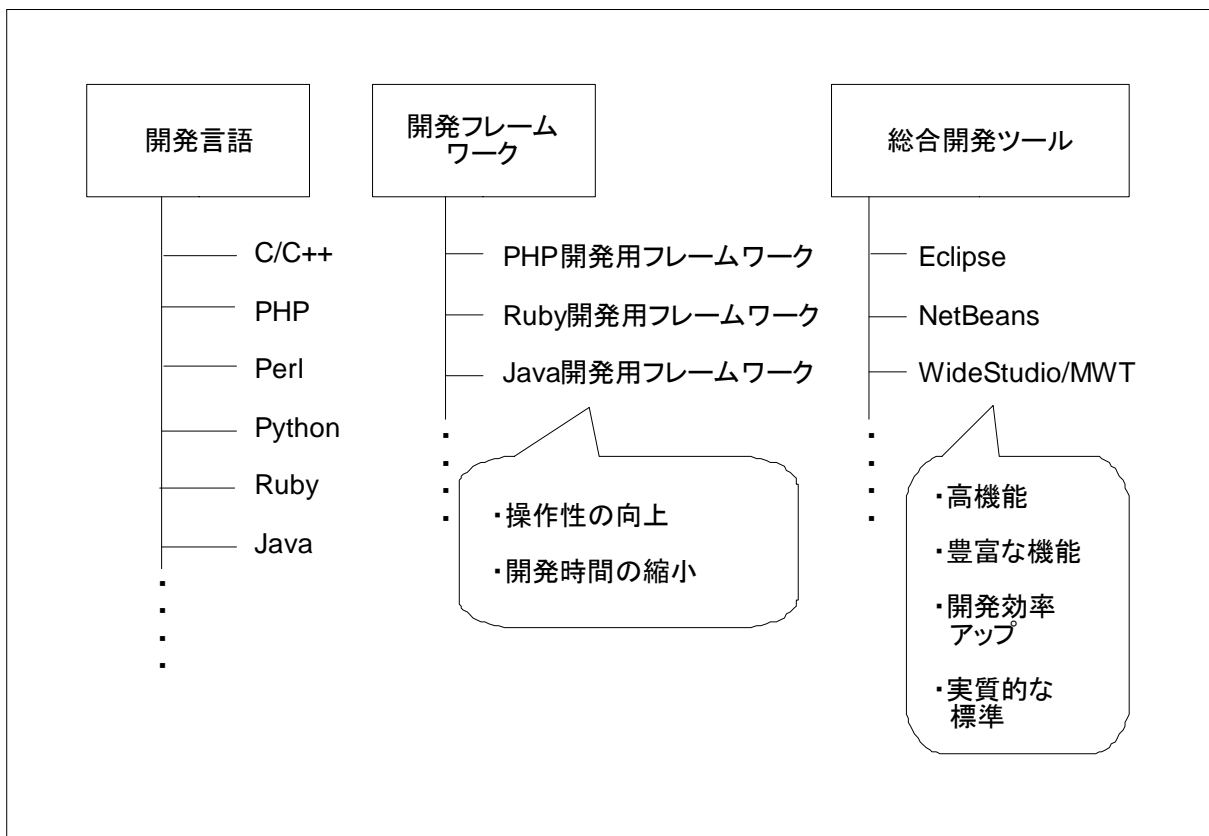


図 I-1-4. 開発言語とフレームワーク

【解説】

1) 主な開発言語

- C 言語 : コンパイラ言語。UNIX の移植性を高めるために開発された。
- C++ : C 言語を拡張してオブジェクト指向化したもの。大規模・複雑な開発を容易とする。
- PHP : 動的に HTML データを生成することが可能なスクリプト言語。
- Perl : インタプリタ形式のプログラミング言語。他の言語の優れた機能を取り入れている。
- Python : スクリプト言語で、手軽に開発を行うことができる。
- Ruby : 本格的なオブジェクト指向スクリプト言語。テキスト処理能力などに優れている。
- Java : オブジェクト指向プログラミング言語。開発と保守の複雑さを低減する。

2) 開発フレームワーク

アプリケーションソフトウェアを開発する際に有用な機能をまとめて提供し、アプリケーションソフトウェアのベースを作成するソフトウェアをフレームワークと呼ぶ。

* PHP 開発用フレームワーク

PHP の開発には、非常に多くの開発用フレームワークが存在する。それぞれが異なる目標を持って作成されているため、選択にあたっては注意が必要である。

* Ruby 開発用フレームワーク

Ruby on Rails というフレームワークに注目が集まっている。

* Java 開発用フレームワーク

(1) 増大する運用管理コストの抑制、(2) アプリケーションの短期開発、(3) 技術者の専門分野に応じた配置といった目的のために、積極的に導入されているフレームワークである。Tomcat や JBoss などのフレームワークは非常に利用者も多い。

3) 統合開発環境

高度なエディタ機能やデバッグ支援機能、バージョン管理機能などを実現する開発環境である。商用ソフトウェアとしても導入実績の高い商品があるが、OSS として提供されている Eclipse、NetBeans、WideStudio などもあり、実質的な標準となっているものもある。

* Eclipse

Java の総合開発環境として標準となっているが、実際は Java、C、C++、Perl、PHP などの多くの言語に対応している。

* NetBeans

NetBeans は、Java 開発用統合開発環境であり、優れた GUI エディタなどの機能を持つ。NetBeans は、サン・マイクロシステムズを中心としたコミュニティにより開発されている。

* WideStudio/MWT

WideStudio/MWT は複数のプラットフォームの GUI アプリケーションを構築するための統合開発環境である。複数の異なるプラットフォームでも動作可能なライブラリを使用しているといった特徴がある。

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-5. OSS のデスクトップアプリケーション	
対応する コースウェア	第5回 (代表的なオープンソースアプリケーション)	

I-1-5. OSS のデスクトップアプリケーション

OSS によるデスクトップアプリケーションの例として、代表的なウィンドウシステム(統合デスクトップ環境)、オフィススイート、ブラウザ、メーラー、グラフィックツールを紹介する。

【学習の要点】

- * デスクトップアプリケーションは、サーバと連携する Web アプリケーションなどとは異なり、コンピュータのデスクトップ上で起動・実行される。
- * 統合デスクトップ環境とは、ユーザにグラフィカルで直感的に理解しやすいインタフェースを提供するソフトウェアである。
- * デスクトップアプリケーションには、ワープロ・表計算などをセットにした OpenOffice.org のように広く利用されているものもある。

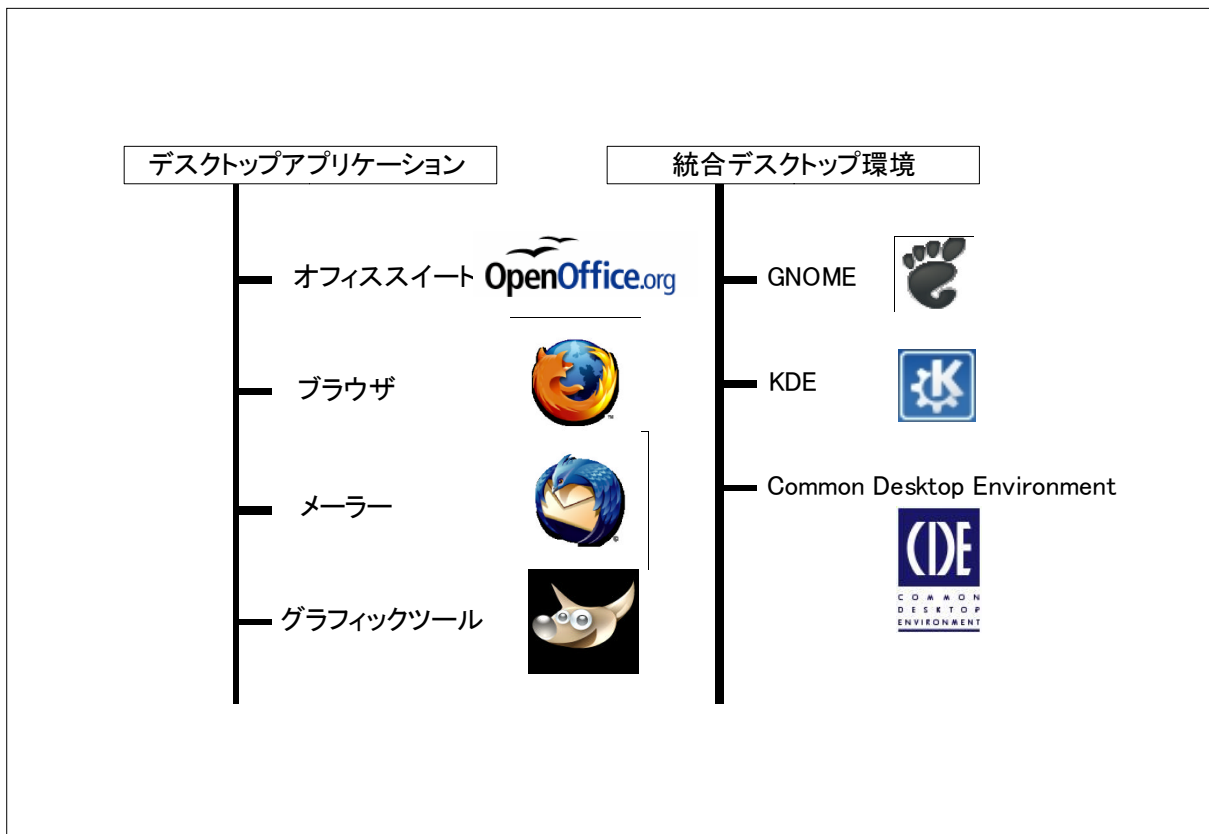


図 I-1-5. デスクトップアプリケーション

【解説】

1) デスクトップアプリケーション

デスクトップアプリケーションとは、コンピュータのデスクトップ上で起動・実行されるアプリケーションプログラムのことである。Linux などのオペレーティングシステムの場合、何らかのサービスを提供するデスクトップアプリケーションと、それらデスクトップアプリケーションが動作するための環境を提供する統合デスクトップ環境がある。

* 統合デスクトップ環境

統合デスクトップ環境とは、コンピュータのグラフィカルユーザインタフェースを提供するソフトウェアのことである。デスクトップ環境は一般にアイコン、ターミナルウィンドウ、ツールバー、メニューバーなどで構成されている。いくつかのデスクトップアプリケーションは、この統合デスクトップ環境の上で実行され、ドラッグ・アンド・ドロップ機能を持つなど、操作性を向上させている。

2) 統合デスクトップ環境

オープンソースソフトウェアとして公開されているデスクトップ環境には、様々なプログラムが提供されている。代表的なものを以下に示す。

* GNOME

* KDE

* Common Desktop Environment

3) デスクトップアプリケーション

コンピュータのデスクトップ上で起動・実行されるプログラムのことである。

* オフィススイート

オフィスツールとは、主にビジネス用途に利用されるワープロ、表計算、描画ツールなどの、各プログラムを連携して利用可能とするアプリケーションである。オープンソースソフトウェアとして、知られている代表的なオフィススイートとして OpenOffice.org がある。

* ブラウザ

Netscape のソースコードを引き継いだ Mozilla Firefox は、タブブラウジング機能やアドオン機能など、先進的な機能を積極的に実現した Web ブラウザである。

* メーラー

Mozilla Thunderbird は、迷惑メールフィルタ機能や RSS リーダー機能を早くから取り入れて、先進的な機能を提供しているアプリケーションである。

* グラフィックツール

グラフィックツールとは、コンピュータグラフィックを描くために必要なデータを作成・保存するためのプログラムである。オープンソースソフトウェアである GIMP は、商用プログラムの替わりとして利用できるほど高機能である。

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-6. OSS のサーバアプリケーション	
対応する コースウェア	第5回 (代表的なオープンソースアプリケーション)	

I-1-6. OSS のサーバアプリケーション

OSS によるサーバアプリケーションとして、最近注目を集めている CRM(顧客管理)、BI(企業内統合データ分析)、ERP(経営資源管理)、CMS(Web サイトのコンテンツ管理)向けのOSSを紹介する。また特定業務向け OSS アプリケーションの例も紹介する。

【学習の要点】

- * 企業の競争力を左右するビジネス・アプリケーション(CRM、BI、ERP)でも、OSS として公開されるものが増えてきている。
- * CMS(Content Management System)では、様々な機能を有するアプリケーションが、OSSとして公開されている。
- * 特に CMS は WEB2.0 の潮流の中で、今後も普及の拡大が予想される。
- * 上記以外にも、様々なサーバアプリケーションがOSSとして公開されている。

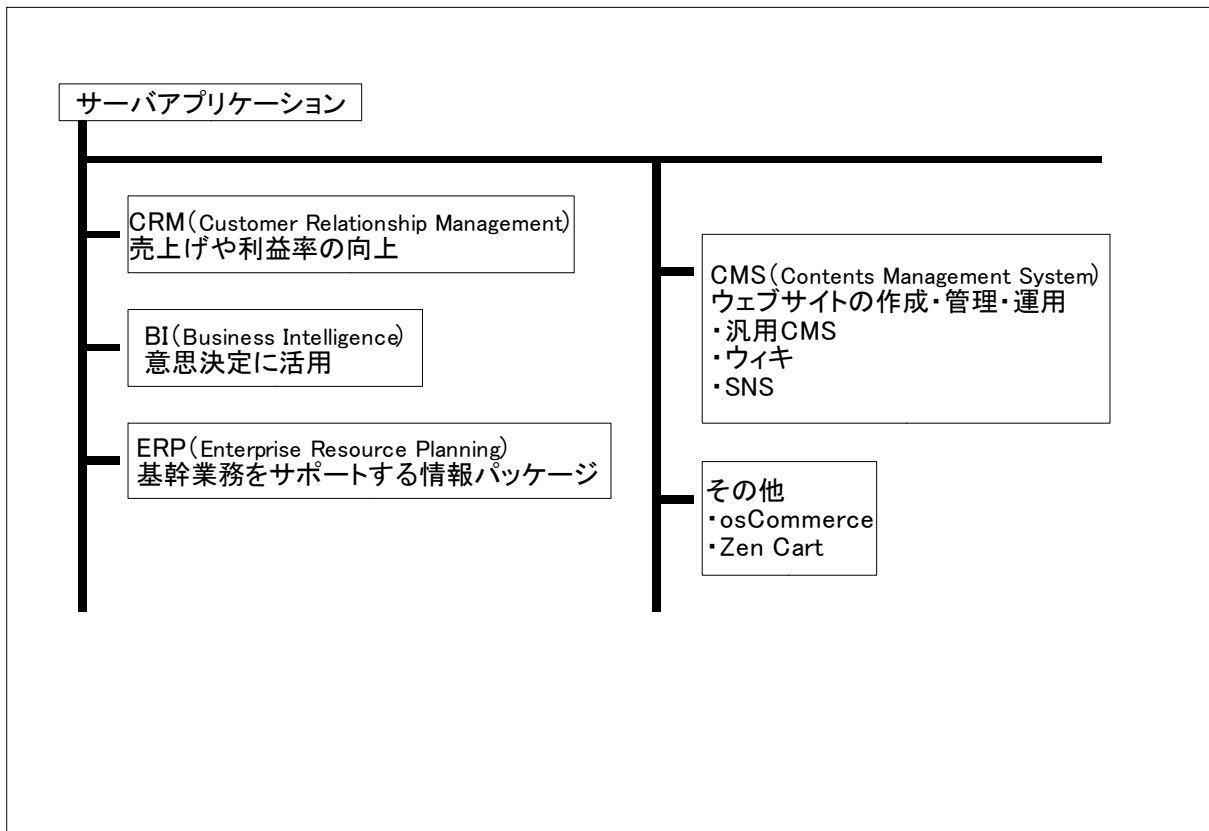


図 I-1-6. OSS のサーバアプリケーション

【解説】

1) 注目を集めるサーバアプリケーション

業務システムなどから蓄積される企業内の膨大なデータの活用など、企業ビジネスにおいて必要となる CRM、BI、ERP 向けの各パッケージが OSS として公開されるようになってきた。

* CRM : Customer Relationship Management

顧客との取引や関係を解析することで、売上や利益率を向上させる仕組みのこと。

- SugarCRM(有償 OSS)、SalesLabor

* BI : Business Intelligence

業務システムや企業内サーバに蓄積されたデータを解析して、経営上の意思決定に利用する仕組み。

- OpenOLAP for MySQL、pentaho

* ERP : Enterprise Resource Planning

受注・販売管理、在庫管理、生産管理といった基幹業務をサポートする情報システムパッケージ。OSS として公開されているものとして、compiere などがある。

2) さらなる市場拡大が期待できる CMS

従来ウェブサイトを作成・管理・運用するためには、HTML ファイルの作成技術や、保存するディレクトリ等の専門知識が必要であった。CMS(Content Management System)では専門的な知識がなくても、ウェブサイトの運用が可能となる。そして、この CMS も OSS のものが増えてきている。

* 汎用 CMS : Geeklog、PHP-Nuke、Typo、XOOPS

一般的なサイトからグループウェア、さらには高機能なポータルサイトまで、様々なサイト構築を可能とする汎用的な CMS

* ブログ : WordPress、Movable Type、Nucleus CMS

複数ブログの管理機能、柔軟なテンプレートエンジン、フィードの提供、トラックバック、アイテム管理などの機能を備えたものが公開されている。

* ウィキ : MediaWiki、PukiWiki、FreeStyleWiki

ウェブブラウザを利用して、ウェブサーバ上の HTML 文書を書き換える機能を提供する。

* SNS : OpenPNE、rktSNS

社会的ネットワークの構築の出来るウェブサイト。ソーシャル・ネットワーキング・サービスまたはソーシャル・ネットワーキング・サイト。

3) その他のサーバアプリケーション

その他にも、様々なサーバアプリケーションが OSS として公開されている

* osCommerce

低コストで高機能なネットショップ EC サイトを構築することを可能とする。

* Zen Cart

osCommerce から派生した EC サイト構築用のアプリケーション。osCommerce と同様、サーチエンジン対策、プロモーション機能、デザインテンプレート機能など、EC サイトを構築するための様々な機能を有している。

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-7. OSS の標準化動向と標準化の意義	
対応する コースウェア	第6回 (オープンソースの市場動向)	

I-1-7. OSS の標準化動向と標準化の意義

OSS の標準化動向を概観する。Linux の標準規格である LSB (Linux Standard Base) や、Java の標準仕様を紹介し、標準を定める意義を解説する。またアジアでの標準化への取り組みや国際標準機関との関係についても触れる。

【学習の要点】

- * LSB により、Linux ディストリビューションの標準仕様が定められており、これによりどのディストリビューションでも備えておく必要のある機能の組み合わせが定められている。
- * Java プラットフォームの標準化仕様は、JCP の標準化プロセスのもとで開発されていて、Java に関係する団体・企業間での利害関係に問題が発生しないようになっている。
- * アジアでも、OSS への期待が高まっている。
- * アジアの中の新興国が OSS を求める理由としては、第一に OSS の低いコストが挙げられることが多い。

LSBとは(Linux Standard Base)
各ディストリビューションが備えておくべき最低限度の機能の組み合わせを定めた標準仕様のこと。同時に標準仕様を決めるプロジェクトの名称。

標準仕様を定める意義
ディストリビューション間の互換性を向上させ、アプリケーションの移植や開発を行ないやすい様にする。

Javaの標準仕様とは
Javaの標準仕様とはJCP (Java Community Process)のもとで開発されている。JCPは、Javaプラットフォームに追加する仕様や技術を、JSRs (Java Specification Requests) に記述している。

図 I-1-7. Linux の標準仕様、Java の標準仕様

【解説】

1) LSB とは

LSB(Linux Standard Base)とは、Linux のどのディストリビューションでも備えておく必要のある機能の組み合わせを定めた標準仕様のことである。またこの仕様を定めるためのプロジェクトの名称でもある。

* ディストリビューション間の互換性

OS の核であるカーネルは、どのディストリビューションにおいてもほぼ共通であるが、その他のアプリケーションについては、ディストリビューションによって異なる組み合わせが選択されている。このため、ディストリビューションごとの互換性は決して高いとはいえない。

* 標準仕様

LSB の目的は、ディストリビューション間の互換性を向上させ、アプリケーションなどの移植および開発を行いやすいように標準的な仕様を定めることである。この仕様は非常に一般的で、かつ著名ないくつかの仕様を取り込み、さらにそれらを拡張しながら作成されてゆく。

- LSB 3.2 : 2007 年 11 月リリース
- LSB4.0 系 : 2008 年リリース予定

2) Java の標準仕様とは

Java プラットフォームの標準化された仕様などは、JCP (Java Community Process) の標準化プロセスのもとで開発されている。この仕様により、Java に関係する団体・企業間での利害関係に問題が発生しないようになっている。

3) アジア地域での取り組みについて

アジアでも、OSS への期待が高まっている。OSS の普及を推進しようとする動きも活発に行われており、例えば日本とアジア各国政府によって開催され、オープンソース普及を目的とした会議にアジア OSS シンポジウムがある。さらには北東アジア OSS 推進フォーラムなども構成されており、中国・韓国の民間企業・研究教育機関と連携体制を築けるようになっている。

* 財団法人 国際情報化協力センター

アジア各国の関係者と連携し、OSS に関連した協力活動を積極的に推進している。これらの活動は、ウェブサイトの充実化を通して、広く一般に公開する活動を行っている。

* アジア OSS シンポジウム

OSS 開発に関する各組織間で、各国／地域での開発支援や利用について情報交換を行い、今後の国際的協力の基盤づくりを目指している。

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-8. OSS の利用状況	
対応する コースウェア	第 6 回 (オープンソースの市場動向) 第 7 回 (オープンソースソフトウェアを用いた事例)	

I-1-8. OSS の利用状況

ビジネスにおける OSS の利用状況を各種調査結果を参照して概説する。Linux および OSS ミドルウェアの市場シェア、企業ユーザの OSS 導入意向やメリット・デメリットを感じる点に触れる。

【学習の要点】

- * すでに OSS は様々なシステム上で、実際に採用・運用されており、安定性やコスト削減等の実績が広く知られるようになってきた。
- * ソースコードを管理することは、サポートサービスを提供する企業やベンダにロックインされることを回避して、長期メンテナンスを実施することが可能である。

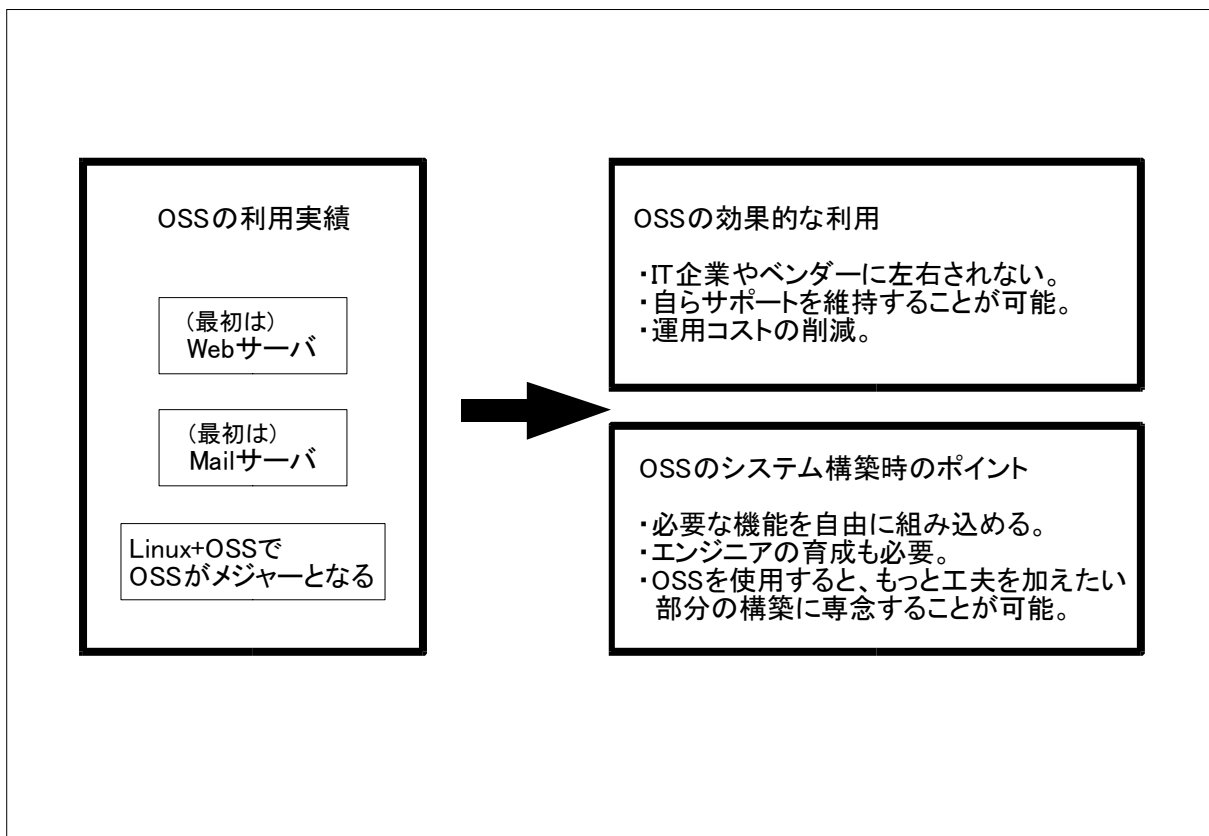


図 I-1-8. OSS の利用実績

【解説】

1) OSS 利用事例

OSS は、すでに様々なシステム上で採用・動作している。どのようなシステムでの採用からはじまり、その後どのようなシステムで動作しているか、どれだけの規模のシステムに対応しているのかを認識することで、OSS で構築されたシステムの信頼性の高さを確認することができる。

* インターネットサーバ

- 最初に利用されはじめたのは、Web サーバやメールサーバ。
- Linux + OSS サーバソフトの組み合わせが OSS をメジャーなものとしてきた。
- ユーザ参加型のインタラクティブな Web システムを支えているのも OSS である。

* OSS が採用されている大規模システムの例

- 15000 ユーザを対象とするメールサーバ
- ページビューが 100 億を超えるウェブサーバ
- 安定性を求められる金融機関システム

2) OSS の効果的な利用

大規模なシステム構築においては、OSS を採用したことにより、運用費用を 3 分の1まで減少させたという事例も報告されている。さらに、Web アプリケーションの構築において、最小限のコストでの開発を実現するためには OSS が最適であると考えられる。OSS 利用の主な利点は以下の通りである。

- * IT企業やベンダの都合に振り回されることがない。(ロックインの回避)
- * ベンダ側のサポート期間とは関係なく、自らサポートを維持することができる。
- * 必要な機能を、自ら組み込むことが可能である。
- * 安定性が高いため、運用コストを削減することが可能である。

3) OSS でのシステム構築時のポイント

ソースコードを解析・研究する技術があれば、自社のシステムに必要な機能を、自由に組み込むことができるが、それを実現するためのエンジニアの育成には、費用や時間も必要となる。すでに公開されている OSS を使用することで、エンジニアはもっとも工夫を加えたい部分の構築に専念することが可能となる。

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-9. OSS による Web システムの構築	
対応する コースウェア	第8回 (オープンソースソフトウェアコミュニティ)	

I-1-9. OSS による Web システムの構築

典型的な Web システムの構築を例に、OSS の使われ方、実際のシステム構築上のポイントやメリット・デメリット、注意点などを説明する。

【学習の要点】

- * Linux のインストールには、ディストリビューションを利用することが便利である。
- * どのディストリビューションを選択したら良いかは、そのサポート内容や管理ツールの充実度で選択すればよい。
- * 常に最新のバージョンを利用するように注意すべきである。
- * ディストリビューションより提供されるパッケージファイルを利用して、簡単に更新処理を行うことができる。

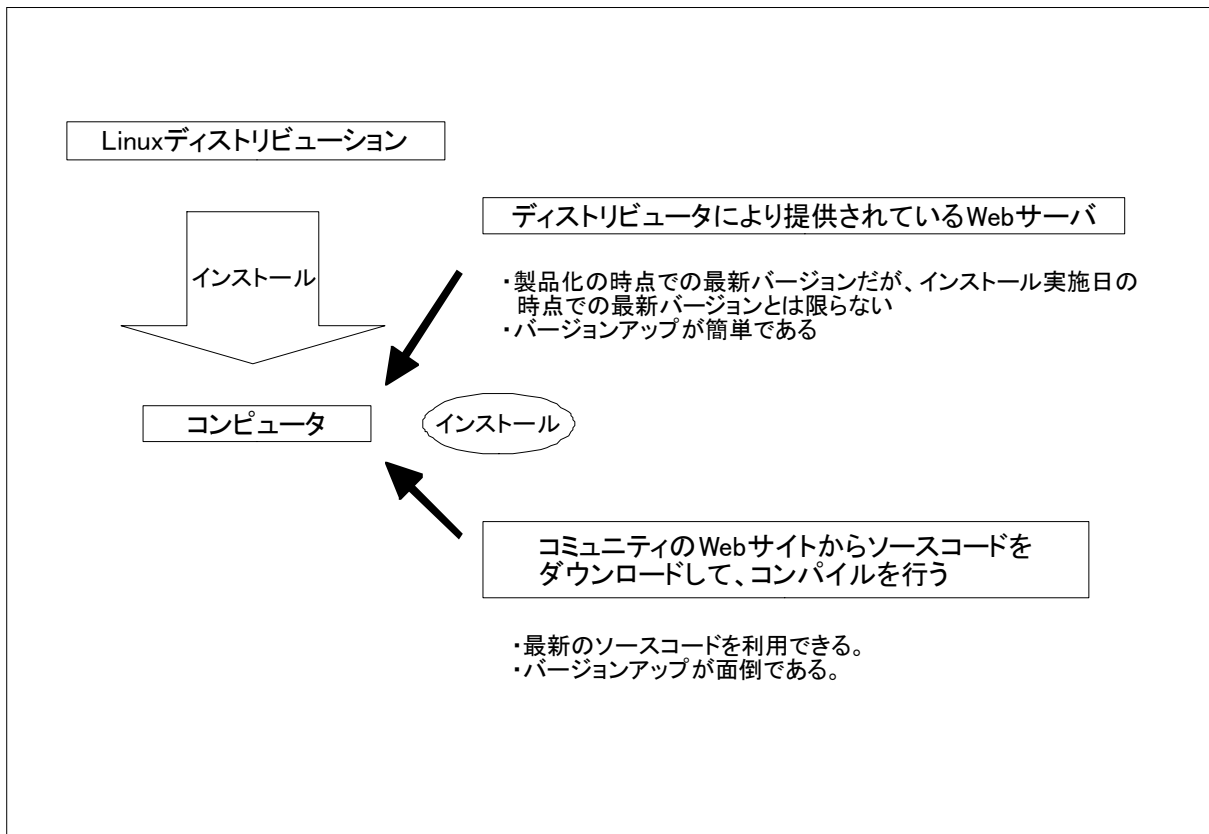


図 I-1-9. Web サーバ構築

【解説】

1) Web システムに必要な OSS

典型的な Web システムであれば、OSS のプラットフォームと Apache などが用意されればよい。

* プラットフォーム(オペレーティング・システム)

OSS のオペレーティング・システムでは、Linux がその代表といえる。ただし、Linux とはカーネルと呼ばれる「核となるプログラム」を示すため、実際にはディストリビューションを利用する方法が簡単である。以下に代表的なディストリビューションを紹介する。

- Red Hat Enterprise Linux : レッドハット社が提供。導入実績が豊富。
- Novell SuSe Enterprise Linux : ノベル社が提供。北欧での導入実績が豊富。
- MIRACLE LINUX : ミラクル・リナックス社が提供。
- Turbolinux : ターボリナックス社が提供
- CentOS : Red Hat Enterprise Linux から商標などを取り除いたもの。
- Fedora : コミュニティベースで開発されている。
- Debian GNU/Linux : コミュニティベースで開発されている。

* Apache HTTP Server

豊富な機能を積極的に取り込んだことにより、非常に高い採用実績を持っている。

- 世界中の Web サーバの 60% が Apache もしくは、その派生製品である。

2) 構築方法

オペレーティング・システムのインストール後に、Apache をインストールして起動させれば、Web サービスの提供は可能となる。

* ソースコードのコンパイル

Apache HTTP Server の開発は Apache Software Foundation が中心となって行われている。Apache のサイト(<http://www.apache.org>)からソースコードをダウンロードして、解凍・展開を行いコンパイルすればよい。

* ディストリビュータより提供されているものを利用

Linux インストール時の選択画面で、Apache をインストールするか選択可能である。また、Red Hat 系と呼ばれるディストリビューションであれば rpm コマンドなど、もしくは yum コマンドなどで簡単に追加インストールを行うことも可能である。

3) メリット・デメリット

著名な OSS では信頼性も高く、ドキュメントも豊富であるが、現状のバージョンでは致命的な不具合が潜んでいる可能性もある(これは商用ソフトでも同じである)。OSS では不具合に対する対応が一般に迅速に行なわれているため、常に最新版を入手するように心がけるべきである。

* ソースコードをコンパイルした場合

最新バージョンが更新された場合に、再ダウンロードとコンパイルが必要となる。

* ディストリビュータから提供されたものを利用した場合

製品化の過程の中で旧バージョンを取り込んでいる場合があるので、最新のバージョンを入手して、更新処理を実施すべきである。この更新処理は非常に簡単である。

スキル区分	OSS モデルカリキュラムの科目	レベル
基礎分野	1 OSS の概要に関する知識 I	基本
習得ポイント	I-1-10. OSS コミュニティの種類と特徴、参加方法	
対応する コースウェア	第8回 (オープンソースソフトウェアコミュニティ)	

I-1-10. OSS コミュニティの種類と特徴、参加方法

OSS コミュニティの代表的な事例を紹介し、OSS コミュニティの種類と特徴を概観する。また OSS コミュニティへの参加方法や参加時の留意点について説明する。

【学習の要点】

- * コミュニティは、ある目的を達成するために、世界中の有志が各人のできることを提供することで成り立っている組織。
- * コミュニティには、開発コミュニティとユーザコミュニティがある。
- * OSS のバグや不足している機能を発見し、その修正点をコミュニティへ送ることで、開発に参加できる。
- * ユーザ会の運営するメーリングリストに登録して技術的な質問を行い、ノウハウを共有することからコミュニティ参加が始まる。

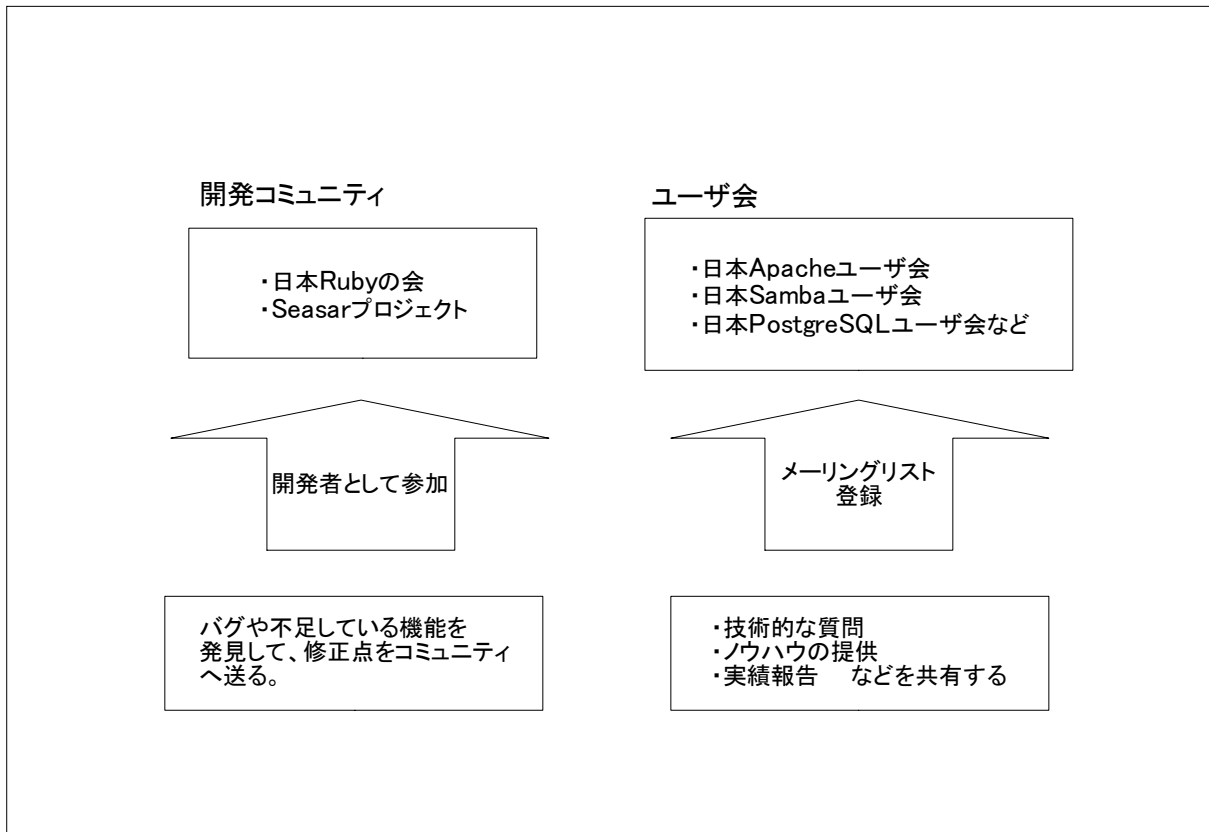


図 I-1-10. OSS コミュニティ

【解説】

1) OSS コミュニティの種類と特徴

OSS コミュニティには、日本人が開発を開始し多くのプログラマが参加しているコミュニティや、日本語ドキュメントの提供やメーリングリストの運営を実施しているコミュニティなどがある。

* 開発コミュニティ

Ruby や Seasar2 は日本人が開発したものである。

- 日本 Ruby の会 : <http://jp.rubyist.net>
- Seasar プロジェクト : <http://www.seasar.org>

* ユーザ会

メーリングリストを運営することで技術的な質問を行い、事例・ノウハウを共有する。

- 日本 Apache ユーザ会 : <http://www.apaceh.jp>
- 日本 Samba ユーザ会 : <http://www.jbug.jp>
- 日本 PostgreSQL ユーザ会 : <http://www.poergresql.jp>
- 日本 PHP ユーザ会 : <http://www.php.gr.jp>
- もじら組 : <http://www.mozilla.gr.jp>
- OpenOffice.org 日本ユーザ会 <http://ja.openoffice.org>

2) 開発コミュニティへの参加方法

開発者のモチベーションは、「自分で作ってみたい」「自分で使ってみよう」といったシンプルなものの。

* コミュニティへの参加方法

OSS にバグや不足していると思われる機能を発見し、その部分の修正を行って、その修正点(パッチ)をコミュニティへ送ることで、開発に参加するケースが多い。

3) 留意点

- * 能力が高く、成果を出せる技術者でないと、積極的にコミットすることができない。
- * 派生ではない本家のソースコードに手を出せる技術者は特定される。