

2007 年度オープンソースソフトウェア活用基盤整備事業
OSS 基本知識の学習ガイダンスに関する調査

OSS 基本知識の学習オリエンテーション

調査報告書

概要版

2008年3月

独立行政法人 情報処理推進機構

目 次

1. 調査の背景と目的.....	3
2. 報告書の構成.....	3
3. OSS 技術の必要性	4
3.1 OSS の利用状況.....	4
3.2 OSS 活用の利点.....	4
3.3 今後の IT システムの方向性と OSS の活用	6
3.4 OSS 技術者へのニーズ・OSS 技術教育へのニーズ.....	8
3.5 技術者として OSS 技術習得に取り組むことの意義.....	9
4. OSS 基本知識との関連を示すための IT 知識体系	13
4.1 知識体系分類軸と IT 知識体系の位置づけ.....	14
4.2 CC2005 の 5 つのカリキュラムの特徴	15
4.3 IT 知識体系	20
4.4 OSS モデルカリキュラム科目のマッピング.....	23
5. IT 知識体系と OSS 基本知識の関連	24
5.1 OSS モデルカリキュラムを基点とした IT 知識体系との関係.....	24
5.2 OSS モデルカリキュラムを習得する上で必要な前提知識.....	29
6. OSS 基本知識として習得すべきポイント	31
7. OSS 基本知識の学習ガイダンス	59

1. 調査の背景と目的

オープンソースソフトウェア（以下、OSS という）の分野では、IT システムを構築するためのオペレーティング・システム、ミドルウェア、ネットワーク、開発ツールが着実に整備されており、情報経済社会を支えるソフトウェア技術基盤としての地位を確立しつつある。情報化社会を支える基盤技術として、OSS 利用ニーズが高まっている一方で、OSS の普及を阻害する大きな要因のひとつとして、OSS を活用できる技術者の人材不足が指摘されている。2007 年に実施された「OSS 技術教育のためのモデルカリキュラムに関する調査」（以下 OSS モデルカリキュラム調査）においても、ユーザ企業や IT サービス事業者における OSS 技術者への期待と実態にギャップが存在することが指摘されている。

OSS がその有効性が社会で広く認知され、普及していくためには、OSS を活用した IT システムの構築・運用・保守を行うことができる、OSS が分かる IT 技術者の育成が重要な課題となっており、OSS 技術教育を行なう際に参照できるモデルカリキュラムの整備ならびに大学、専門学校、IT 研修機関等への普及が必要である。

以上のような背景のもと、本調査事業では、OSS モデルカリキュラム調査の成果を基に、基本レベルに相当する 26 科目（OSS 基本知識）の受講者及び講師を対象として、OSS 技術教育のオリエンテーションで活用することを想定した「OSS 基本知識の学習体系」ならびに各科目の学習を通じて確実に習得することが期待される根幹となる知識・概念を解説した学習ガイダンスを策定した。

2. 報告書の構成

本報告書の構成は以下の通りである。

(1) OSS 基本知識の学習体系

OSS 基本知識に関する学習体系に関して、以下に事項を整理した。

- ・ OSS 技術の必要性 (第 3 章)

OSS モデルカリキュラム調査ほか、各種資料等で得られている OSS 技術または OSS 技術者へのニーズを整理して提示すると共に、技術者として OSS 技術の習得に取り組むことの意義を整理した。

- ・ OSS 基本知識との関連を示すための IT 知識体系 (第 4 章)

IEEE¹, ACM²による Computing Curricula 2005（以下、CC2005）をベースに、OSS 基本知識を相対的に関連付けるための IT 知識体系を抽出した。

- ・ IT 知識体系と OSS 基本知識の関連(第 5 章)

上記で定義された IT 知識体系と OSS 基本知識の各科目間の関係を明らかにし、可視化することによりその関連性を明確化した。また、全体的な関連性を俯瞰図として表すことにより、OSS 基本知識の IT 知識体系における位置づけ、また逆に IT 知識体系の OSS 基本知識体系における位置づけを明らかにした。

- ・ OSS 基本知識として習得すべきポイント(第 6 章)

OSS 基本知識 26 科目ごとに 10 項目の習得すべきポイント（学習を終了した時点で習得して

¹ The Institute of Electrical and Electronics Engineers

² Association for Computing Machinery

いることが期待される概念または手法) を明らかにした。

(2) OSS 基本知識の学習ガイダンスの策定 (別冊 [26 分冊])

基本レベルに相当する 26 科目の各習得ポイントに関する学習の要点、解説図および解説をまとめた。

3. OSS 技術の必要性

3.1 OSS の利用状況

Linux や Apache (HTTP server)、Sendmail 等に代表されるオープンソースソフトウェア (以下 OSS) は、その設計図にあたるソースコードが公開されており、誰でも無償での使用や改良、ならびに再配布が行えるソフトウェアである。OSS は主にインターネットを支えるサーバソフトウェアとして、あるいはアカデミックな用途を中心に普及していったが、2001 年の Linux 2.4 のリリースを大きな契機として、エンタープライズ市場、とりわけ企業の根幹となる基幹システム市場への浸透が進展することとなった。Linux 2.4 で行われた数多くの機能拡張や性能向上に加え、IBM や国内大手ベンダが Linux 対応を相次いで打ち出したことも OSS がエンタープライズ市場へ浸透した大きな要因としてあげることができる。インプレスと矢野経済研究所が 2006 年に出版した「Linux オープンソース白書 2006」によれば、企業や公共団体の情報システムにおいて Linux サーバを導入している割合は 28% になり、商用 UNIX サーバの 26% を上回る結果となった。Linux の導入状況は業種によって大きく異なっているが、特に、政府系機関や公共団体での導入率は高く、8 割を超えている。一方、導入が遅れていた金融業でも、最近では、システム統合のためのプラットフォームとして Linux サーバを導入する動きも出てきており、金融システムのようなミッションクリティカルな分野においても、導入コストや信頼性、また拡張可能性をその理由として導入が進みつつある。

さらに、近年では、基本ソフトやインターネットサーバソフトウェアだけでなく、データベースや企業内におけるコンテンツ管理、情報共有環境、アプリケーションフレームワークといった、従来商用ソフトウェアが中心であった領域においても OSS の利点が明らかになるとともに、その利用が進展しつつある。

3.2 OSS 活用の利点

OSS を活用する利点として、まず第一に「経済性」を挙げることができる。実際、前述の「Linux オープンソース白書 2006」をはじめ、多くの調査でも初期導入コストが安いという点が導入時の最も大きな理由となっている。しかし、OSS を活用することには、経済性以外にもさまざまなメリットがある。以下に OSS 活用の利点として重要と思われるポイントを挙げる。

(1) 経済性

OSS の導入において、初期のライセンス料が不要というのは明確なメリットである。もっとも、導入後の保守サポートまで含めた TCO (Total Cost of Ownership) で考えた場合には、必ずしも商用ソフトに比べて圧倒的に優位とは言い切れない部分がある。

ただし、Amazon.com や Intel では、自社のインフラを Linux に移行することによって、年間数千万ドル以上のコストが削減できたとされているように、大規模なインフラ構築においては基本ソフトウ

ウェアやミドルウェアのライセンス料は膨大であり、OSS のスケールメリットは無視できない。

また、OSS のアプリケーションを導入することにより、スクラッチから開発するのに比べ、工期削減、ひいてはコスト削減の効果も期待できる。

(2) ベンダロックインの回避

OSS 導入のメリットとして中長期的に重要な意味を持つのが、特定ベンダへの依存回避である。企業が特定ベンダの製品、あるいはデータ形式に依存することは、バージョンアップや機能追加に関する裁量権をほぼベンダ側に委ねることを意味しており、価格面のみならず、将来的に柔軟な情報システム運用を考えた際には、大きな障壁となる。

一方、OSS ではソースコードが公開されているために、特定のベンダに限定されることなく、複数のベンダが同じ土俵で調達競争に参加できる。また、システムの改良や更新の際にも、他のベンダの参入余地が残されている。従来、商用ソフトウェアを導入する場合には、ユーザ企業にとっては多少高いコストを払っても信頼を買うという意味合いもあったが、OSS の信頼性が向上するとともに、大手ベンダが OSS サポートをビジネスとして展開している昨今においては、特定のベンダにロックインされる理由が無くなってきている。

(3) 相互運用性の確保

欧州連合では、各国の電子政府の連携を図る eEurope 計画において、OSS の利用が推奨されている。複数の組織における情報システムの連携を考える場合には、相互運用性の確保が極めて重要な課題となるが、それぞれが異なる商用ソフトウェアを利用している場合には互いにブラックボックスの状態となり、相互接続の構築ならびに検証には多大な期間、工数を要することになる。一方、OSS を活用する場合には、ソースコードを互いに入手し精査することによって互いに検証可能な環境を提供することができる。

国内においても、近年企業の合併・再編が進み、各社が持つ基幹システムの統合運用が重要な課題となっているが、最近ではこうしたシステム統合、相互運用性を鑑みて、金融など、信頼性の観点を特に重視する業種においてさえ、オープンなプラットフォームに基づくシステム構築が注目されるようになってきている。

(4) 技術の蓄積

IT ベンダの視点から見ると、OSS を導入することは社内的な技術を蓄積する上での大きな意味を持っている。商用ソフトウェアの組み合わせによるシステム構築では、ソフトウェアの機能そのものへの理解よりもそれをどうやって使うのかといった表層的な部分の知識が蓄積されることとなり、ソフトウェア開発そのものの技術を養っていくことは困難である。また、受託システム開発においては、ソフトウェア開発のスキルは必要にはなるものの、開発されたプログラムの中身が評価にさらされることは少なく、また、他の技術者による優れたコーディング技術を学習する機会も少ない。

一方、OSS をベースにした開発では、すでに開発されている優れたソフトウェアを実際に目にすることができるよう、開発したソフトウェア自体も OSS として公開する場合には、多くの開発者の目に触れることとなり、技術的な切磋琢磨が行われることが期待できる。

ベンダの OSS 技術レベルを判断する上で、対象分野における OSS の中心的開発者となるようなエンジニアがいるかどうか、世界と競える技術者を抱えているかどうかは IT ベンダにとって、将来的な大きな資産となるであろう。

3.3 今後の IT システムの方向性と OSS の活用

今後の IT システムの方向性を考えたとき、OSS の活用はさらにその重要性を増すことが想定される。具体的には、今後の IT システムの方向性として、以下が考えられる。

- オープンなプラットフォームでのシステムやサービスの連携（オープン化）
システムやサービスの連携ニーズが高まるにつれ、各サブシステムの仕様やインタフェースの仕様が目に見えるオープンなプラットフォーム上での開発ニーズが高まっており、OSS を活用したオープンな開発プラットフォームを活用する流れが生まれてきている。
- 個人や中小企業のインフラやサービス基盤としての進展（コモディティ化）
ブログや SNS をはじめ、Web2.0 と呼ばれる流れにおいては、従来情報発信力を持たなかった個人や中小企業が手軽に情報発信やサービスを展開できるようになる。安価に導入でき、かつ機能的にも商用ソフトウェアに引けをとらない Wiki（文書共有編集システム）や CMS（コンテンツ管理システム）等の OSS はこうした流れを牽引するポテンシャルを持っている。
- IT を用いたサービス化に伴うコストや開発期間の短縮（サービス化）
サービスの多様化に伴い、サービス基盤のスケラビリティ、工期の短縮（スピード化）、コストの低減が求められるようになってきている。Amazon.com や google の例をひくまでもなく、膨大な量のサービスをロバストに提供していく枠組みとして、Linux 等の OSS をベースにすることは、サービス創出のスピードアップ、コスト削減に大きくつながっていくと期待されている。

以上の IT システムの今後の方向性、ならびに OSS 活用の関係を示したものが図 1 である。

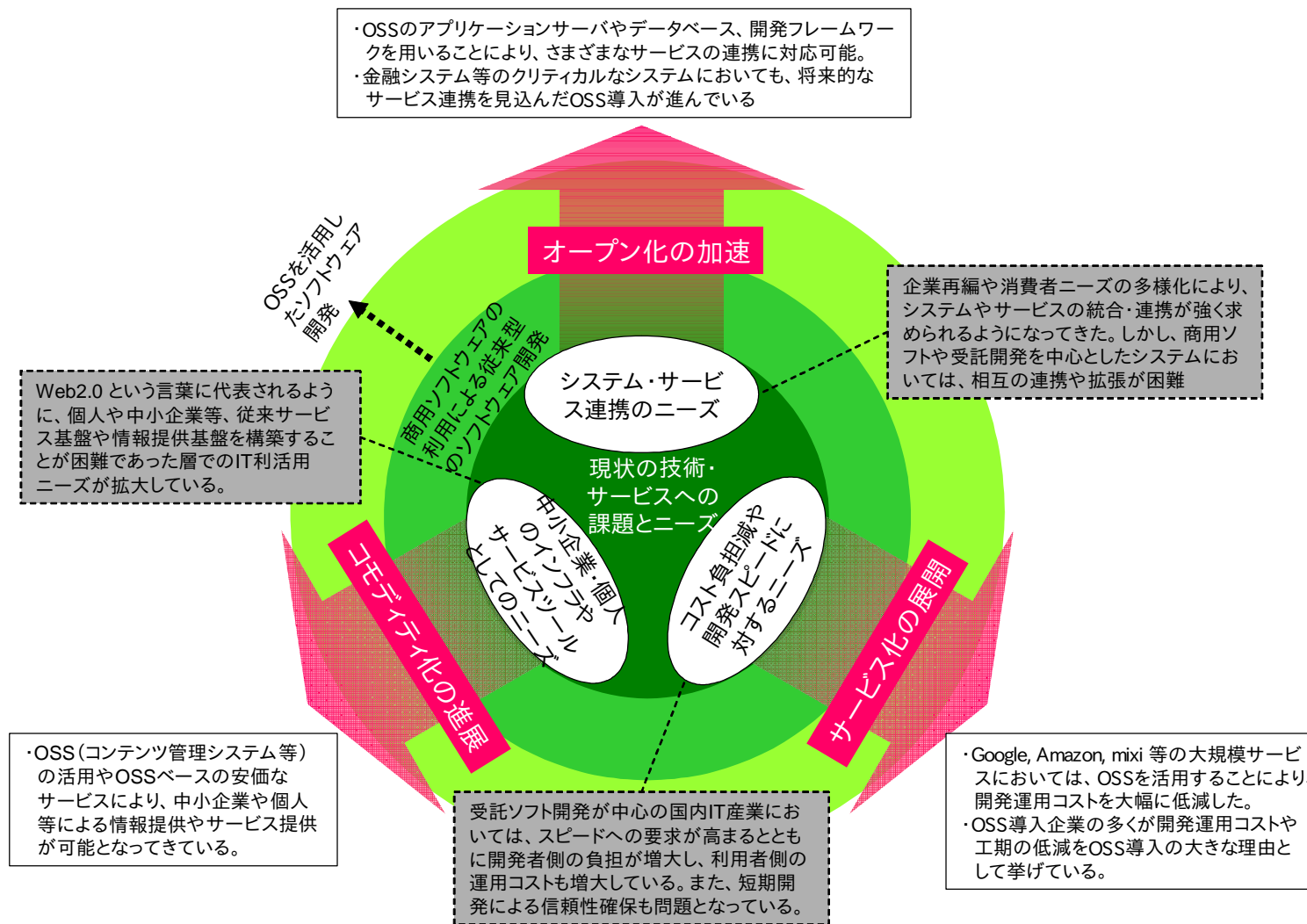
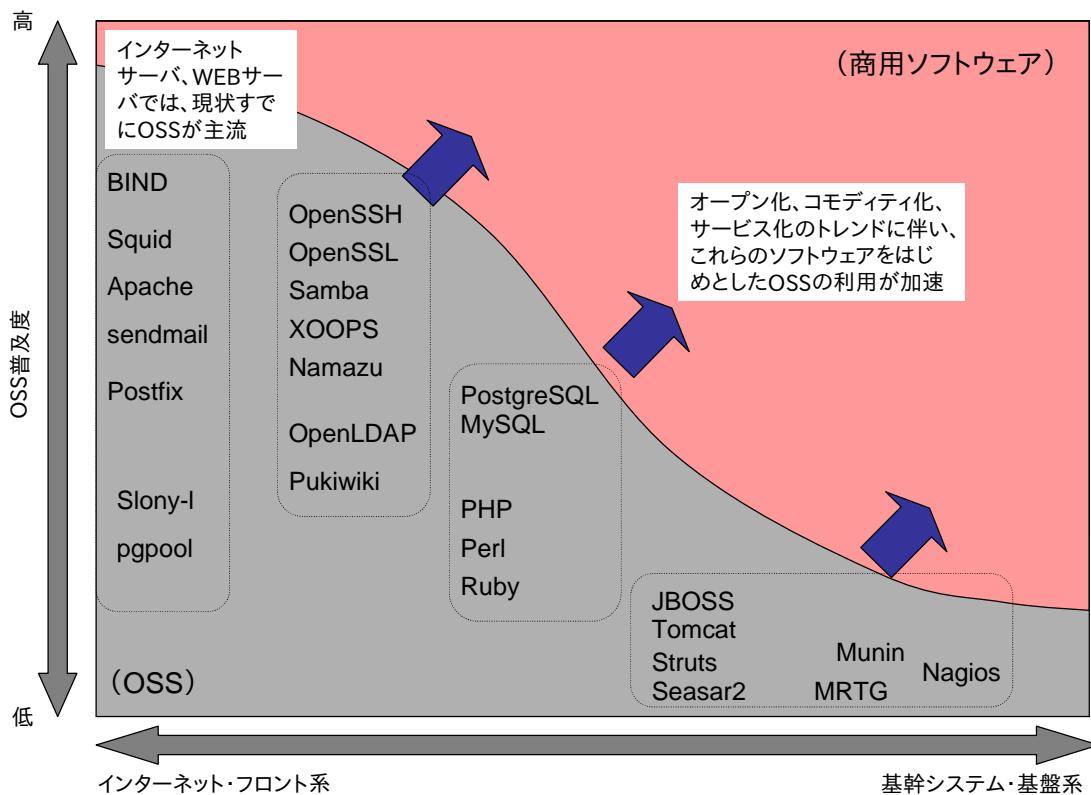


図 1 IT 技術の方向性と OSS

また、図2はOSSの普及度とITシステムの方向性を示したものである。Webサーバをはじめとするインターネットサーバでは既にOSSが主流となっている。一方、情報共有や認証、データベース、アプリケーションサーバにおいても、近年商用ソフトウェアにとって代わることが可能なほどOSSが成熟してきている。また、業務サーバや基幹サーバのようなクリティカルなシステムにおいてもOSS活用事例が出てきている。オープン化、コモディティ化、サービス化のトレンドに伴い、図2に示したようなソフトウェアを中心に従来商用ソフトウェアが中心であった分野においてさらにOSS活用が進むものと見られる。



(「オープンソースで構築！ITシステム導入虎の巻」(IPA OSSセンター 2007年) 図1-4をもとに作成)

図2 OSSの普及度とITシステムの方向性

3.4 OSS技術者へのニーズ・OSS技術教育へのニーズ

以上のように、OSSの活用が進む中で、OSS普及の重要な問題点として「人材不足」を上げる声が多い。

インプレス、矢野経済研究所が2006年に発表した「Linux オープンソース白書2006」によれば、今後のOSS活用の問題点として、ベンダのサポートや安定性や信頼性を依然不安視する声があるものの、圧倒的に多いのは人材不足を指摘する声であり、7割強がOSSの問題点として、管理やスキルのある人材がいないことを挙げている。これはOSSそのものの普及を阻害する要因となっている可能性があると同時に、上述のようにOSS活用による「技術力の蓄積」という観点から考えれば、国内ITベンダーの技術力低下を招きかねない状況であるともいえる。

また、2004年に実施された「オープンソース/フリーソフトウェア開発者オンライン調査」（日本版 FLOSS-JP 調査）では、OSS 開発者に焦点をあてているとはいえ、OSS 開発開始年齢でいえば欧米よりも3~4才高く、参加年数は1年半短いという結果が出ている。情報系の学生の比率が低いといった特徴も挙げられており、大学等における専門課程の中で体系的に OSS 教育が行われているわけではなく、多くは独学で学んでいるという状況を垣間見ることができる。企業内においても、OJT や自己研鑽が主たる OSS 技術の習得方法となっており、社内外の研修等による体系的学習は行われていない。特に今後より重要となると思われる開発環境や開発フレームワーク等に関しては、社内教育体制が整備されておらず、今後の OSS 利用ニーズの高まりに相反して、教育体制の不備が普及の阻害要因となることが懸念される。

以上の状況から考えて、大学・専修学校、研修機関、企業等において、体系的な OSS 学習、特に、既存の IT 知識や IT 教育カリキュラムと関連付けられた OSS 学習体系の整備が急務であるといえる。

3.5 技術者として OSS 技術習得に取り組むことの意義

ベンダやユーザ企業における OSS 技術者の利用ニーズに加え、IT 技術者側から見た視点においても、OSS 技術を体系的に習得する意義は大きい。

図3は OSS モデルカリキュラムを、CC2005 において提示されたカリキュラムの特徴空間軸にしたがってマッピングしたものである（詳細は4章で議論する）。この図で示されるように、OSS モデルカリキュラムは、CC2005 でカバーされる領域の中で、組織や管理に関する知識領域、コンピュータ科学の原理や理論に関する領域、ならびにハードウェアの設計や製造に関する技術領域を除いた部分を主にカバーしており、企業等において必要とされる実践的な IT 技術者に共通して求められる IT 知識体系と密接に関連している（本報告書では、この領域を「IT 知識体系」と呼ぶ。詳細は4章参照）。

また、図4は IT 知識体系で学習される内容と OSS 基本知識体系で学習される内容の関連の概略を示したものである（IT 知識体系と OSS 基本知識体系の関連付けの詳細については5章を参照）。

IT 技術者が、より実践的な技術を習得するためには、知識のみならずそれらを実際に問題解決に結び付けていくための技量を実習により補うことが必要である。OSS 技術の習得は、知識に加えて技量を身につけるための実践的な場を提供するものである。

IT 技術者として OSS 技術習得に取り組むことの意義は、以下のようにまとめることができる。

- ソフトウェア開発における実践的な技術の習得
特定の商用ソフトウェアを利用できるといったレベルではなく、いろいろな要素技術を自分の手で組み合わせたり、ソースコードも含めて学習することにより、原理・理論をプロダクトレベルで結び付けて理解できるとともに、適用事例や研究開発事例の成果を利用することにより、ソフトウェア開発の現場で適用可能な実践的な技術を養うことができる。
- 問題解決能力の向上
商用ソフトウェアではある問題に対する解決方法を、ベンダが体系的に提供し、それ以外の方法は推奨されない。一方、OSS では一つの問題に対して複数の解決方法が存在するのが普通であり、それらをインターネット上から探し出し、それらを比較検証して最適な問題解決の方法を検討する。これは一見回り道のようにも見えるが、IT 技術者として真に必要な問題解決能力を向上させることができる。
- 技術革新への適応力の向上

特定の商用製品のみを扱うのとは異なり、OSSではさまざまな選択肢を常に比較検討すること、また、たゆみのない機能追加や機能変更に追従することを通じて、技術の進展や変化に適応する能力を養うことができる。

- 生産性と品質の向上

多様な OSS を組み合わせて、迅速にシステム開発を行うことによって、生産性を大きく向上させることができる。また、品質の高い OSS を再利用することによって、新規開発部分を最小限に抑えることができ、結果的に品質を向上することにもつながる。さらに、優れた OSS のソースコードは最高の教材であり、それを真似ることによって、高品質のソフトウェアを作成するスキルの向上にもつながる。

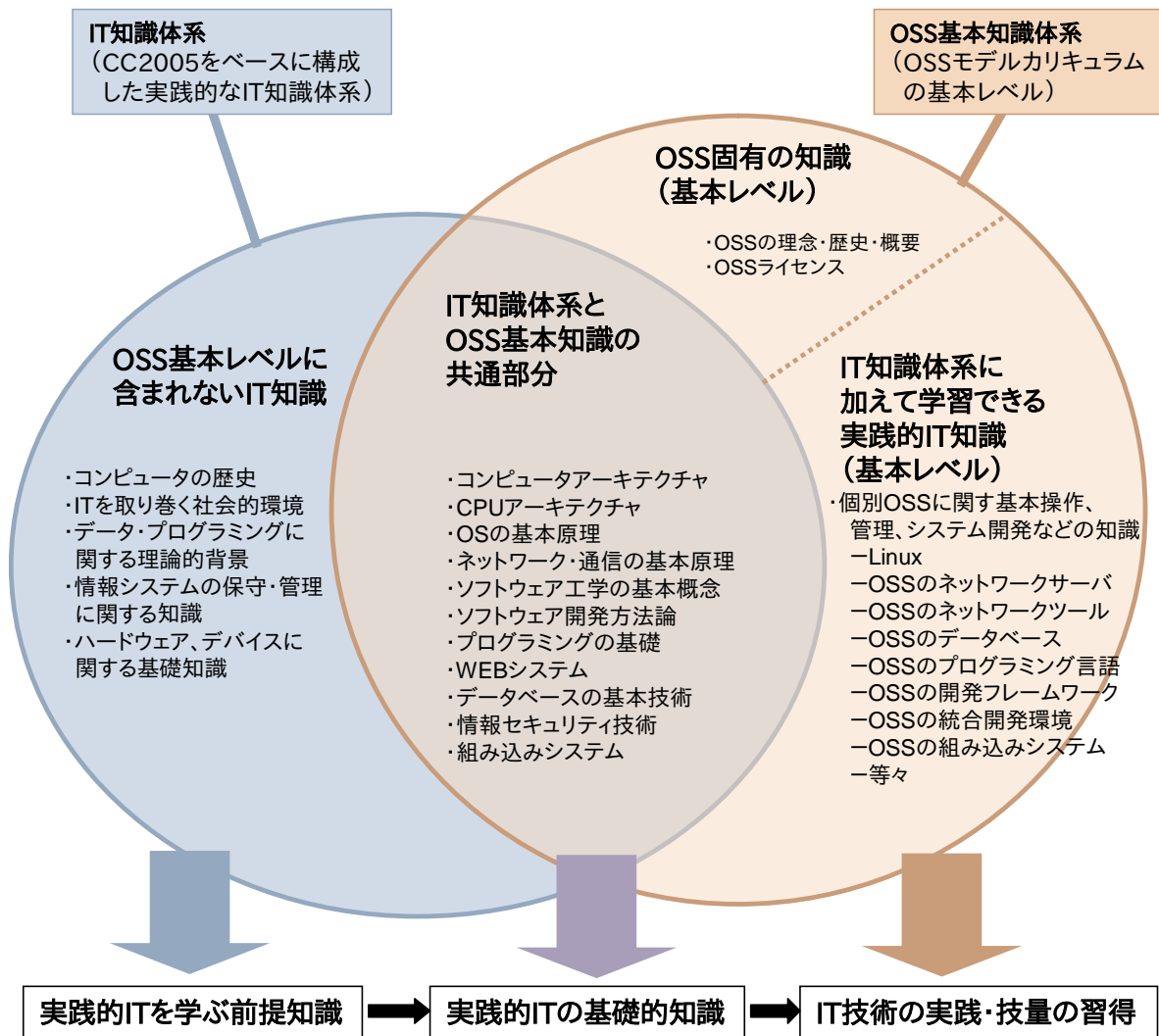


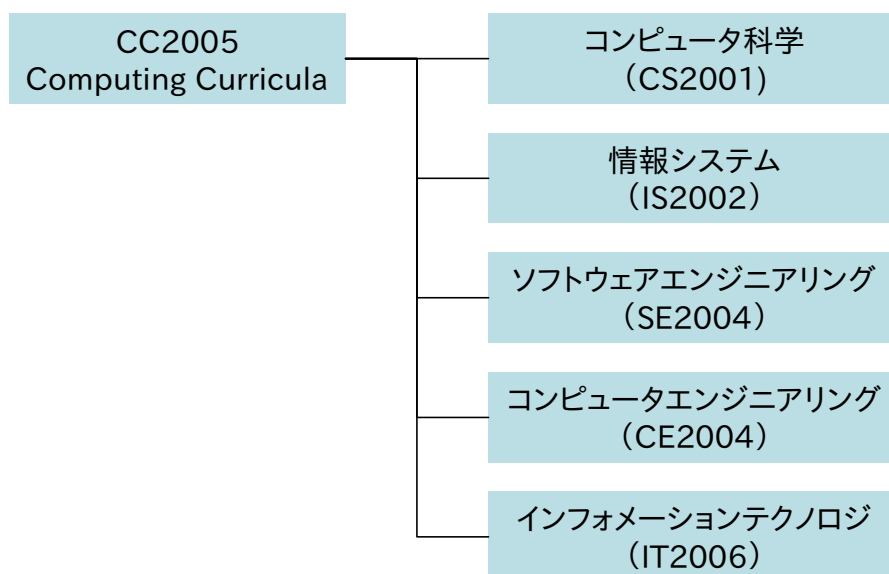
図4 IT知識体系とOSS基本知識の関連

4. OSS 基本知識との関連を示すための IT 知識体系

OSS モデルカリキュラムを活用した実際の教育においては、大学、専門学校、企業の IT 研修等における既存の IT 教育プログラムに対して、OSS モデルカリキュラムから必要な科目を選択し、OSS 技術を含めた IT 教育カリキュラムを設定することが想定されている。

このためには、OSS モデルカリキュラムと IT 知識体系の科目毎の関連、ならびに全体像としての関連を明らかにすることが必要である。本章では、この目的のために必要となる IT 知識体系を整理する。

対象とする IT 知識体系としては、現状広く教育現場で受け入れられている、もしくは今後受け入れられる可能性があるもので、かつ国際的な普遍性を考慮したものであることが望ましい。そこで、IT 分野で国際的な標準カリキュラムとして認知度が高い ACM³、AIS⁴、IEEE-CS⁵による CC2005 (Computing Curricula 2005 図 5 参照) をベースに整理を行なった。また、CC2001、CC2005 をベースに情報処理学会を中心として検討が進められている J07 も参考にした。



[Computing Curricula 2005, The Overview Report をもとに作成]

図 5 CC2005 の構成

IT 知識体系の抽出手順は以下の通りである。

- (1) CC2005 に含まれる 5 つの領域 (IT, CS, IS, SE, CE) のそれぞれに対して、各領域を構成する科目を特徴付けるために、CC2005 (Computing Curricula 2005 – The Overview Report, Sep. 30, 2005) で定義されたカリキュラムの特徴空間 (problem space of computing) の分析軸にしたがってマッピングを行なった。
- (2) マッピング後、各領域間で内容的に重複する科目を抽出した。また、重複する科目においては、IT

³ The Association for Computing Machinery (ACM)

⁴ The Association for Information Systems (AIS)

⁵ The Computer Society (IEEE-CS)

知識体系として代表させる科目を選択した。以上のプロセスにより、IT 知識体系とその周辺領域を構成する科目を整理した。

また、OSS モデルカリキュラムの基本レベルの 26 科目についても同様にマッピングを行い、本特徴空間における OSS モデルカリキュラムと IT 知識体系との全般的関連性を整理した。

4.1 知識体系分類軸と IT 知識体系の位置づけ

CC2005 においては、5 つの各カリキュラムを特徴付けるために図 6 に示した分類軸が設定されている。横軸は、理論的から応用的に至る軸であり、大学、研究機関等における新しい原理や革新的方法論の教育においては、左側に位置される教育カリキュラムが、逆に、より実践的なアプローチにより企業・社会における問題解決を担う人材教育においては、右側に位置される教育カリキュラムが選択されることになる。

また、縦軸はハードウェア・アーキテクチャから組織的事項や情報システムに至る軸であり、デバイス設計やコンピュータ内部のハードウェア技術の教育においては、下側に位置される教育カリキュラムが、逆に、人や組織と情報システムの関連、情報技術が社会に与える影響の分析手法の教育においては、上側に位置される教育カリキュラムが選択されるべきである。

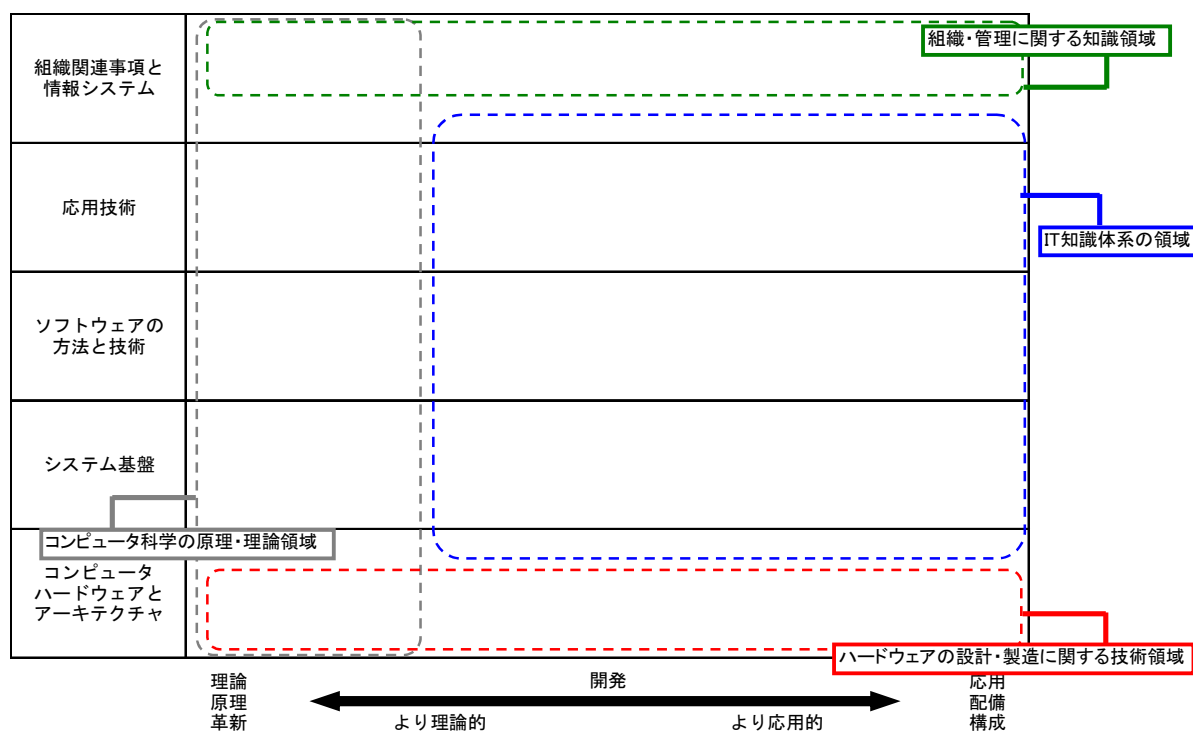


図 6 CC2005 の特徴空間

[CC2005 The Overview Report (pp.16, Figure 2.2) , J07 中間報告等をもとに作成]

ここで「IT 知識体系」の定義は、「産業界等の実践的な場における情報システムの基本的な設計、開発、ならびに利用において必要とされる基礎的知識」とし、特徴空間上(図 6) 破線で囲んだ領域を「IT 知識体系」のカバーする領域とした。

これに対し、「組織・管理に関する知識領域」は、情報システムの組織における活用や管理において必

(5) CC2005 のソフトウェアエンジニアリング (SE) 領域のマッピング

CC2005 の SE 領域の各科目を特徴空間上にマッピングした結果を図 11 に示す。図で示されたように、CC2005 の SE 領域は、情報システムの信頼性が求められる状況を背景として、質の高いソフトウェアを生み出すために必要なソフトウェアエンジニアリング技術を中心としたカリキュラムとなっている。

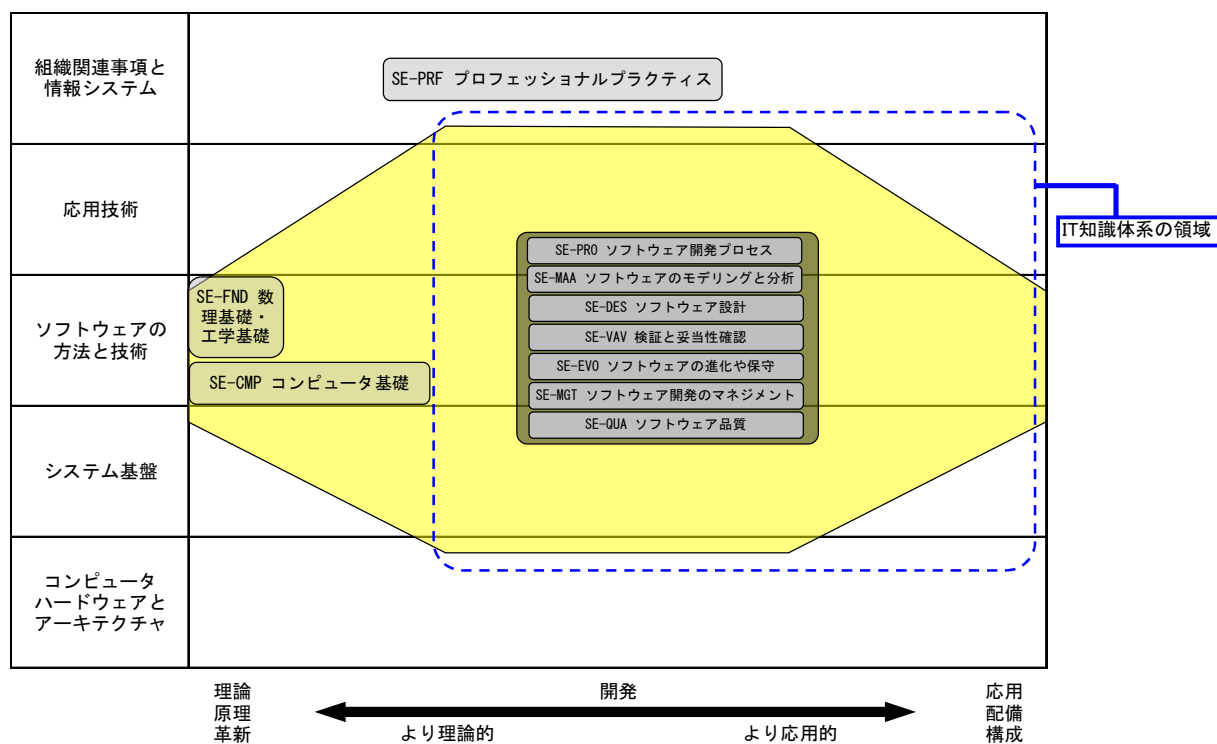


図 11 CC2005 のソフトウェアエンジニアリング(SE) 領域科目の特徴

4.3 IT 知識体系

各科目のマッピング結果をもとに内容的に重複する科目を検討し、最終的に構成した IT 知識体系を図 12 に示す。また、IT 知識体系に含まれる科目の概要ならびに構成を表 1 に示す。

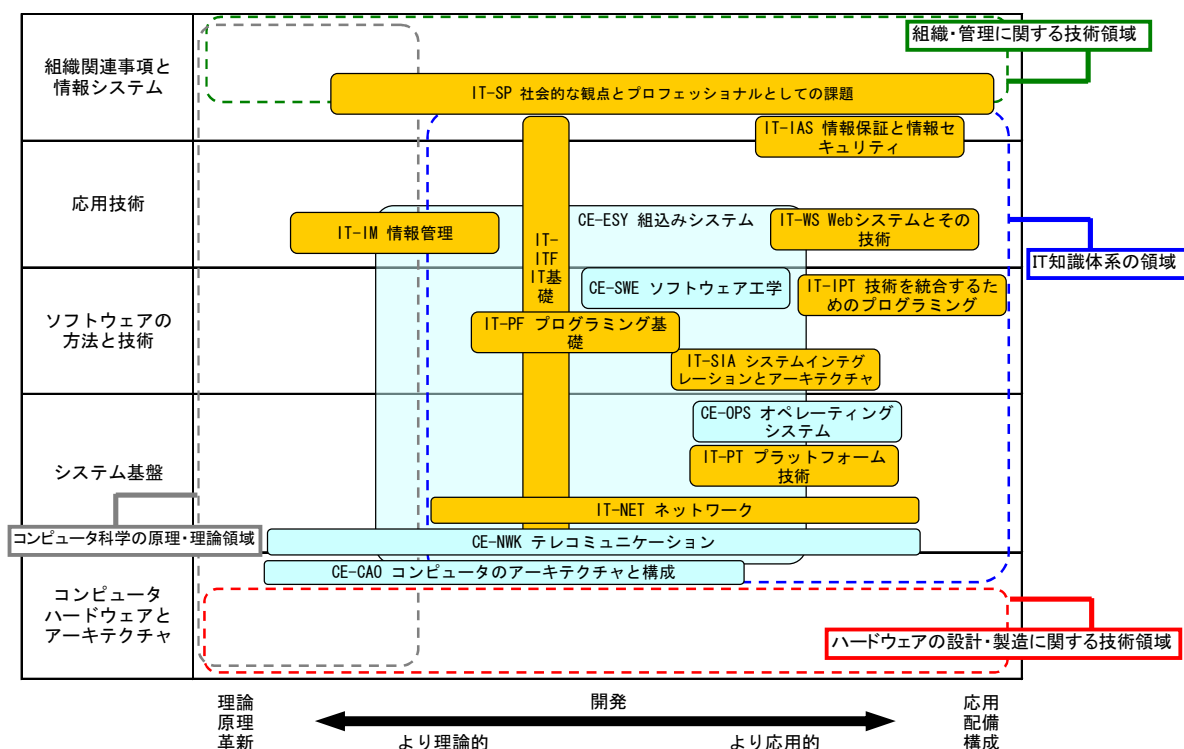


図 12 CC2005 から抽出した IT 知識体系

ここで策定した IT 知識体系は、4.1 で述べたように、実践的な IT 技術の習得に重点が置かれたものである。IT 知識体系の周辺知識として、以下の領域を設定した。

- ・ 組織・管理に関する技術領域
- ・ コンピュータ科学の原理・理論領域
- ・ ハードウェアの設計・製造に関する技術領域

「組織・管理に関する技術領域」では、意思決定の仕組みやコミュニケーション技術、法的側面や情報システムの社会的側面が含まれる。これらは主に CC2005 の情報システム (IS) 領域に含まれる科目から構成される。IT 知識体系領域の習得を踏まえ、より上流の設計や問題解決・判断を行なうためには、この領域の習得が必要になる。

「コンピュータ科学の原理・理論領域」では、計算原理やアルゴリズム、ソフトウェアや応用技術のための基礎理論が含まれる。これらは主に、CC2005 のコンピュータ科学 (CS) 領域に含まれる科目から構成される。これらの理論的知識については、IT 知識体系領域の習得により概略を習得することが可能であるが、さらに研究開発的要素を必要とする IT 技術者においては、この領域の習得が必要になる。

「ハードウェアの設計・製造に関する技術領域」では、コンピュータアーキテクチャや設計・製造技

術が含まれる。これらは主に、CC2005 のコンピュータ工学（CE）領域に含まれる科目から構成される。IT 知識体系領域の習得を踏まえ、組み込みシステム等のハードウェア技術との接点でのシステム開発業務を担うには、この領域の習得が必要になる。

表 1 IT 知識体系の科目一覧

分野	科目名	概要	目標	1	2	3	4	5	6	7	8	9	10	11	12	13	
組織関連事項と情報システム	1	IT-IAS 情報保証と情報セキュリティ	運用上の問題、ポリシー、攻撃と防御の方法、リスク分析、情報セキュリティなど	システムを安全かつ確実に運用・管理するために不可欠なセキュリティの考え方と実現方法を理解する。	IT-IAS1. 基礎的な問題	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. ポリシー	IT-IAS5. 攻撃	IT-IAS6. 情報セキュリティ分野	IT-IAS7. フォレンジック(情報証拠)	IT-IAS8. 情報の状態	IT-IAS9. 情報セキュリティサービス	IT-IAS10. 脅威分析モデル	IT-IAS11. 脆弱性		
	2	IT-SP 社会的な観点とプロフェッショナルとしての課題	コンピュータの歴史、社会性、専門性、倫理的な問題と責任、法的な問題	ITを有効に活用するのに必要な人的・社会的知識を修得する。	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. 組織の中の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	データや情報に関する収集、編成、利用、管理、モデリング、変換、表現、信頼性、セキュリティ	情報やデータの収集、設計、管理、利用方法を修得する。	IT-IM1. 情報管理の概念と基礎	IT-IM2. データベース問合わせ言語	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	Web上のアプリケーションやソフトウェア、デジタルメディアの融合、社会的問題、倫理的な問題、安全性	Webシステムの仕組みと、その構成要素、開発方法について学ぶ。	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	基本的プログラミング概念、イベント駆動プログラミング、オブジェクト指向プログラミング、基本データ構造、アルゴリズムのプロセス	プログラミングの概念と基本技術を修得する。	IT-PF1. 基本データ構造	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-IPT 技術を統合するためのプログラミング	種々のプログラミング言語の特徴と利用方法、アーキテクチャ、アプリケーションプログラミングインタフェース、プログラミングの実践	複数のシステムやアプリケーションの通信や協調を実現するためのプログラミング方法について学習する。	IT-IPT1. システム間通信	IT-IPT2. データ割り当てと交換	IT-IPT3. 統合的コーディング	IT-IPT4. スクリプティング手法	IT-IPT5. ソフトウェアセキュリティの実現	IT-IPT6. 種々の問題	IT-IPT7. プログラミング言語の概要						
	7	CE-SWE ソフトウェア工学	要求分析と仕様決定、設計、構築、テスト、運用、保守、プロジェクト管理	高品質のソフトウェアを効果的かつ効率的に構築するための方法や知識を修得する。	CE-SWE0. 歴史と概要	CE-SWE1. ソフトウェアプロセス	CE-SWE2. ソフトウェアの要求と仕様	CE-SWE3. ソフトウェアの設計	CE-SWE4. ソフトウェアのテストと検証	CE-SWE5. ソフトウェアの保守	CE-SWE6. ソフトウェア開発・保守ツールと環境	CE-SWE7. ソフトウェアプロジェクト管理	CE-SWE8. 言語翻訳	CE-SWE9. ソフトウェアのフォールトトレランス	CE-SWE10. ソフトウェアの構成管理	CE-SWE11. 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	要求収集、調達、構築、プロジェクト管理、検証、ITアプリケーションと組織プロセスの相互関係	システムを具現化するための手順や方法について学ぶ。	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	データ通信、電気通信、インターネット・イントラネット、インフラセキュリティ、ネットワークアプリケーションなど	ネットワークの基礎として、通信技術、ネットワークアプリケーション、セキュリティなどについて学ぶ。	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	CE-NWK テレコミュニケーション	LANとWAN、クライアントサーバー、データの整合性、セキュリティなど	コンピュータネットワークの概要を把握し、通信の仕組みを理解する。	CE-NWK0. 歴史と概要	CE-NWK1. 通信ネットワークのアーキテクチャ	CE-NWK2. 通信ネットワークのプロトコル	CE-NWK3. LANとWAN	CE-NWK4. クライアントサーバーコンピューティング	CE-NWK5. データのセキュリティと整合性	CE-NWK6. ワイヤレスコンピューティングとモバイルコンピューティング	CE-NWK7. データ通信	CE-NWK8. 組込み機器向けネットワーク	CE-NWK9. 通信技術とネットワーク概要	CE-NWK10. 性能評価	CE-NWK11. ネットワーク管理	CE-NWK12. 圧縮と伸張
	11	IT-PT プラットフォーム技術	ハードウェア・ソフトウェアの基礎、ITシステム構成要素の統合方法	ITシステムを構成するハードウェア、ソフトウェアの基礎を修得する。	IT-PT1. オペレーティングシステム	IT-PT2. アーキテクチャと機構	IT-PT3. コンピュータインフラストラクチャ	IT-PT4. デプロイメントソフトウェア	IT-PT5. ファームウェア	IT-PT6. ハードウェア							
	12	CE-OPS オペレーティングシステム	オペレーティングシステムの基本原理と目的、利用法、設計と実装	オペレーティングシステムの原理と目的を理解し、その利用法や設計・実装時の要点を修得する。	CE-OPS0. 歴史と概要	CE-OPS1. 並行性	CE-OPS2. スケジューリングとディスパッチ	CE-OPS3. メモリ管理	CE-OPS4. セキュリティと保護	CE-OPS5. ファイル管理	CE-OPS6. リアルタイムOS	CE-OPS7. OSの概要	CE-OPS8. 設計の原則	CE-OPS9. デバイス管理	CE-OPS10. システム性能評価		
ウェアラブルデバイス	13	CE-CAO コンピュータのアーキテクチャと構成	コンピュータアーキテクチャの概要、基本原理	コンピュータアーキテクチャの概要を把握し、コンピュータの原理と扱い方を修得する。	CE-CA00. 歴史と概要	CE-CA01. コンピュータアーキテクチャの基礎	CE-CA02. メモリシステムの構成とアーキテクチャ	CE-CA03. インタフェースと通信	CE-CA04. デバイスサブシステム	CE-CA05. CPUアーキテクチャ	CE-CA06. 性能・コスト評価	CE-CA07. 分散・並列処理	CE-CA08. コンピュータによる計算	CE-CA09. 性能向上			
複数領域にまたがるもの	14	IT-ITF IT基礎	IT分野の概要と他のコンピュータ関連分野との関連	ITの背景や特徴を理解する。	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(学科)とそれに関連のある分野(学科)	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	CE-ESY 組込みシステム	マイクロコントローラ、組込みプログラム、リアルタイムオペレーティングシステム等の構成要素、および、設計手法、開発環境、開発プロセスについて学習する。	組込みシステムの開発に必要な、マイクロコントローラ、組込みプログラム、リアルタイムオペレーティングシステム等の構成要素、および、設計手法、開発環境、開発プロセスについて学習する。	CE-ESY0. 歴史と概要	CE-ESY1. 低電力コンピューティング	CE-ESY2. 高信頼性システム的设计	CE-ESY3. 組込み用アーキテクチャ	CE-ESY4. 開発環境	CE-ESY5. ライフサイクル	CE-ESY6. 要件分析	CE-ESY7. 仕様定義	CE-ESY8. 構造設計	CE-ESY9. テスト	CE-ESY10. プロジェクト管理	CE-ESY11. 並行設計(ハードウェア、ソフトウェア)	CE-ESY12. 実装
					CE-ESY13. リアルタイムシステム設計	CE-ESY14. 組込みマイクロコントローラ	CE-ESY15. 組込みプログラム	CE-ESY16. 設計手法	CE-ESY17. ツールによるサポート	CE-ESY18. ネットワーク型組込みシステム	CE-ESY19. インタフェースシステムと混合信号システム	CE-ESY20. センサ技術	CE-ESY21. デバイスドライバ	CE-ESY22. メンテナンス	CE-ESY23. 専門システム	CE-ESY24. 信頼性とフォールトトレランス	

4.4 OSS モデルカリキュラム科目のマッピング

OSS モデルカリキュラムについて、CC2005 各カリキュラムと同様に特徴空間にマッピングを行なった結果を図 13 に示す。OSS モデルカリキュラムは、CC2005 から抽出した IT 知識体系の領域とほぼ一致しており、CC2005 の五分野との比較において、より実践的な技術に重きが置かれていることが分かる。

IT 知識体系で習得する技術と OSS モデルカリキュラムで習得する技術の間には密接な関連があることが本図からも示唆されるが、次章ではこの関連性をさらに詳細に分析することとする。

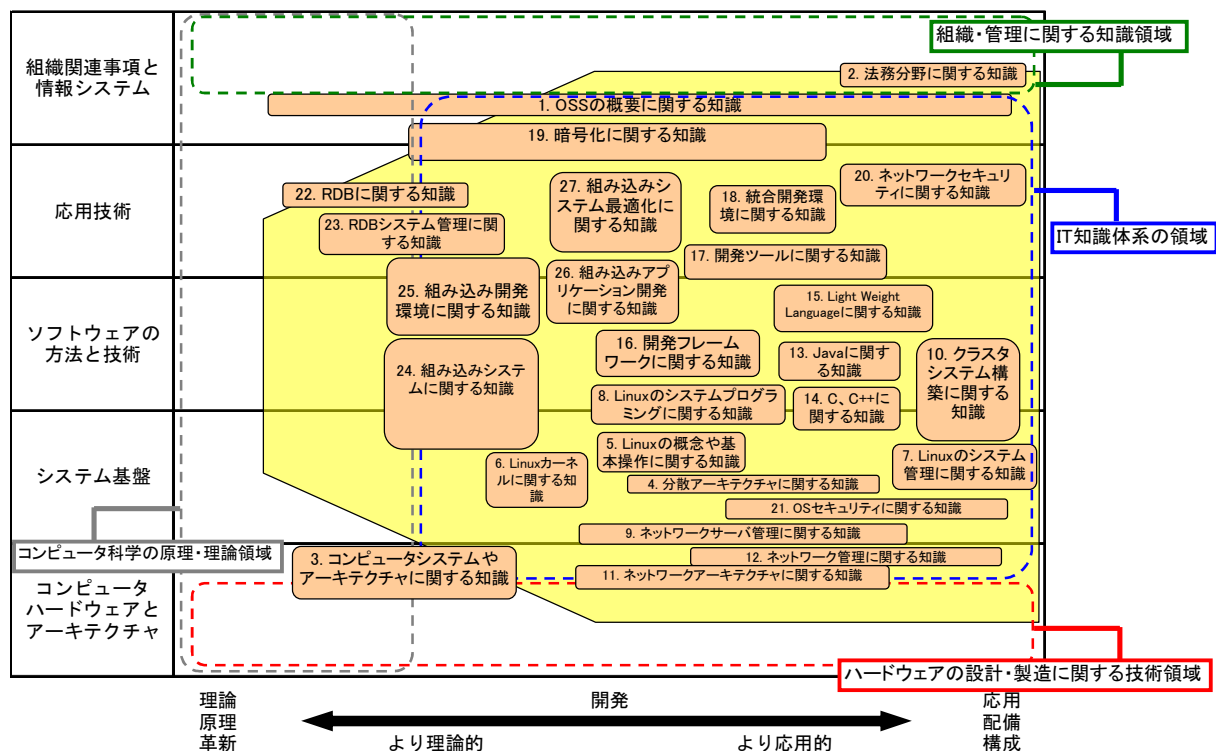


図 13 OSS モデルカリキュラムの位置付け (図 3 再掲)

5. IT 知識体系と OSS 基本知識の関連

4 章で整理した IT 知識体系と OSS 基本知識 (OSS モデルカリキュラムのうち、基本レベルに相当する 26 科目) について、科目間の関連性を検討し、IT 知識体系と OSS 基本知識の全体的な関連性を可視化する。

5.1 OSS モデルカリキュラムを基点とした IT 知識体系との関係

OSS 基本知識全体に対する IT 知識体系上の関連部分を表 2 に示す。表中網掛けの部分は OSS 基本知識との対応関係がある科目を表す。また、網掛けされた科目において、[]内には関連する OSS 基礎知識を示した。

ここで示されるように、OSS モデルカリキュラムと IT 知識体系の間には密接な関連があり、OSS モデルカリキュラムを学習することにより、実践的な IT 技術で裏付けられた IT 知識をより深く理解することができる。

網掛けされていない部分については、OSS 基礎知識ではカバーされない知識領域であるが、大きく以下の 2 つに分けることができる。

- ・ コンピュータの歴史や IT を取り巻く社会的環境、IT の基本的な概念・知識など IT (ならびに OSS) を学ぶ上での基本的前提となるもの
- ・ OSS 応用知識 (OSS モデルカリキュラムの応用レベル) でカバーされると想定されるもの

また、IT 知識体系で学習できない OSS モデルカリキュラム固有の知識については、表 3 に一覧を示した (網掛け部分は IT 知識体系によりカバーされる部分を表す)。OSS モデルカリキュラム固有の知識には、OSS 固有の知識である OSS の概念やライセンスに加えて、IT 知識体系に加えて学習できる実践的 IT 知識がある。後者は、Linux に関する知識 (システム管理やネットワーク、システムプログラミング等)、OSS のデータベースやネットワークサーバ、OSS の開発フレームワーク、OSS の統合開発環境に関する知識などが含まれる。

基本的な IT 技術を習得した技術者においては、ここで示された OSS モデルカリキュラム固有の知識のみを習得することにより、OSS 基本知識全体を習得することができるとともに、OSS を利用した実践的な演習を行なうことにより、知識をより実践的な技量・スキルへ成熟させていくことができる。

表 2 OSS 基本知識と関連付けられる IT 知識体系の全体像

分野	科目名	1	2	3	4	5	6	7	8	9	10	11	12	13	
組織関連事項と情報システム	1 IT-IAS 情報保証と情報セキュリティ	IT-IAS1. 基礎的な問題 [19-1-1] [20-1-1]	IT-IAS2. 情報セキュリティの仕組み(対策) [19-1-5, 8] [20-1-1, 7]	IT-IAS3. 運用上の問題 [20-1-1]	IT-IAS4. ポリシー [20-1-1]	IT-IAS5. 攻撃 [20-1-2, 3, 4, 5]	IT-IAS6. 情報セキュリティ分野 [7-1-1] [20-1-7]	IT-IAS7. フォレンジック(情報証拠) [19-1-8] [21-1-2]	IT-IAS8. 情報の状態	IT-IAS9. 情報セキュリティサービス [20-1-1] [21-1-2]	IT-IAS10. 脅威分析モデル	IT-IAS11. 脆弱性 [19-1-4] [20-1-3, 4]			
	2 IT-SP 社会的な観点とプロフェッショナルとしての課題	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 [3-1-6]	IT-SP4. チームワーク	IT-SP5. 知的財産権 [2-1-5, 6]	IT-SP6. コンピュータの法的問題 [20-1-1]	IT-SP7. 組織の中のIT	IT-SP8. プロフェッショナルとしての倫理的問題と責任	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-IM1. 情報管理の概念と基礎 [3-1-8] [22-1-1] [23-1-5]	IT-IM2. データベース問合わせ言語 [22-1-1]	IT-IM3. データアーキテクチャ [22-1-2, 4, 8]	IT-IM4. データモデリングとデータベース設計 [22-1-5, 6, 7] [23-1-3]	IT-IM5. データと情報の管理 [13-1-6, 7] [23-1-1, 2, 4, 5]	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 [1-1-7] [13-1-1, 6]	IT-WS2. 情報アーキテクチャ [1-1-7] [16-1-6]	IT-WS3. デジタルメディア	IT-WS4. Web開発 [3-1-8]	IT-WS5. 脆弱性 [19-1-5, 8] [20-1-1, 5] [21-1-8]	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. 基本データ構造 [14-1-3, 5, 6, 9]	IT-PF2. プログラミングの基本的構成要素 [17-1-3, 5]	IT-PF3. オブジェクト指向プログラミング [13-1-3, 8] [15-1-6]	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-IPT 技術を統合するためのプログラミング	IT-IPT1. システム間通信 [1-1-3]	IT-IPT2. データ割り当てと交換	IT-IPT3. 統合的コーディング [13-1-3]	IT-IPT4. スクリプティング手法 [15-1-1, 2, 3, 4, 5]	IT-IPT5. ソフトウェアセキュリティの実現 [19-1-2]	IT-IPT6. 種々の問題	IT-IPT7. プログラミング言語の概要 [14-1-1, 2] [15-1-1] [17-1-3]							
	7 CE-SWE ソフトウェア工学	CE-SWE0. 歴史と概要 [17-1-1]	CE-SWE1. ソフトウェアプロセス	CE-SWE2. ソフトウェアの要求と仕様 [16-1-6]	CE-SWE3. ソフトウェアの設計 [15-1-6] [16-1-6]	CE-SWE4. ソフトウェアのテストと検証	CE-SWE5. ソフトウェアの保守	CE-SWE6. ソフトウェア開発・保守ツールと環境 [1-1-4] [18-1-1, 2]	CE-SWE7. ソフトウェアプロジェクト管理	CE-SWE8. 言語翻訳 [3-1-6]	CE-SWE9. ソフトウェアのフォールトトレランス	CE-SWE10. ソフトウェアの構成管理 [7-1-1] [17-1-4]	CE-SWE11. ソフトウェアの標準化 [1-1-6]		
	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 [1-1-4] [3-1-6]	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 [3-1-1] [11-1-1, 2, 3, 4, 6]	IT-NET2. ルーティングとスイッチング [11-1-8, 9]	IT-NET3. 物理層 [11-1-2, 5, 6]	IT-NET4. セキュリティ [19-1-1, 2, 3, 4, 7, 8, 9, 10, 11] [20-1-7, 8] [21-1-1, 3, 4]	IT-NET5. アプリケーション分野 [1-1-5] [3-1-8]	IT-NET6. ネットワーク管理 [11-1-5] [12-1-1, 2] [19-1-7] [20-1-7]								
	10 CE-NWK テレコミュニケーション	CE-NWK0. 歴史と概要 [3-1-1] [11-1-1]	CE-NWK1. 通信ネットワークのアーキテクチャ [3-1-1] [11-1-2]	CE-NWK2. 通信ネットワークのプロトコル [7-1-8] [9-1-3, 4] [11-1-3, 6, 7, 8, 9, 10] [12-1-5]	CE-NWK3. LANとWAN [11-1-4, 6]	CE-NWK4. クライアントサーバコンピューティング [1-1-3] [9-1-1] [15-1-2]	CE-NWK5. データのセキュリティと整合性 [9-1-4] [19-1-2, 3, 4, 8] [20-1-7, 8] [21-1-3, 4, 7]	CE-NWK6. ワイヤレスコンピューティングとモバイルコンピューティング	CE-NWK7. データ通信	CE-NWK8. 組み込み機器向けネットワーク [24-1-2]	CE-NWK9. 通信技術とネットワーク概要 [11-1-5, 9]	CE-NWK10. 性能評価	CE-NWK11. ネットワーク管理 [12-1-1, 2] [20-1-7, 8] [21-1-4]	CE-NWK12. 圧縮と伸張	
		CE-NWK13. クラスタシステム [10-1-1]	CE-NWK14. インターネットアプリケーション [1-1-5, 7]	CE-NWK15. 次世代インターネット	CE-NWK16. 放送										
	11 IT-PT プラットフォーム技術	IT-PT1. オペレーティングシステム [1-1-3] [3-1-7] [6-1-1, 2] [24-1-1]	IT-PT2. アーキテクチャと機構 [3-1-5]	IT-PT3. コンピュータインフラストラクチャ	IT-PT4. デプロイメントソフトウェア [1-1-4] [12-1-2]	IT-PT5. ファームウェア	IT-PT6. ハードウェア								
	12 CE-OPS オペレーティングシステム	CE-OPS0. 歴史と概要 [3-1-7] [5-1-1] [6-1-2] [8-1-4]	CE-OPS1. 並行性 [6-1-1, 2, 4] [26-1-1, 2, 4, 5]	CE-OPS2. スケジューリングとディスパッチ [3-1-7] [6-1-2, 3, 4, 5]	CE-OPS3. メモリ管理 [3-1-7] [8-1-7]	CE-OPS4. セキュリティと保護 [19-1-4, 8] [21-1-1]	CE-OPS5. ファイル管理	CE-OPS6. リアルタイムOS [6-1-2, 5] [24-1-6, 7, 8] [26-1-3, 4, 5, 8]	CE-OPS7. OSの概要	CE-OPS8. 設計の原則 [6-1-1, 3] [12-1-1, 2]	CE-OPS9. デバイス管理	CE-OPS10. システム性能評価 [21-1-1]			
13 CE-CAO コンピュータのアーキテクチャと構成	CE-CAO0. 歴史と概要 [3-1-1, 2]	CE-CAO1. コンピュータアーキテクチャの基礎 [3-1-1, 2, 4] [24-1-5]	CE-CAO2. メモリシステムの構成とアーキテクチャ	CE-CAO3. インタフェースと通信 [3-1-5] [26-1-3, 7, 8] [27-1-2]	CE-CAO4. デバイスサブシステム	CE-CAO5. CPUアーキテクチャ [3-1-3]	CE-CAO6. 性能・コスト評価	CE-CAO7. 分散・並列処理 [10-1-5] [26-1-6] [27-1-1, 2]	CE-CAO8. コンピュータによる計算	CE-CAO9. 性能向上					
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 [1-1-4] [20-1-1]	IT-ITF2. 組織の問題	IT-ITF3. ITの歴史	IT-ITF4. IT分野(学科)とそれに関連のある分野(学科)	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 CE-ESY 組み込みシステム	CE-ESY0. 歴史と概要 [24-1-1, 2]	CE-ESY1. 低電力コンピューティング [24-1-3] [27-1-2]	CE-ESY2. 高信頼性システムの設計	CE-ESY3. 組み込み用アーキテクチャ [24-1-1, 3, 4] [27-1-1]	CE-ESY4. 開発環境 [25-1-1, 2, 3, 4] [27-1-3]	CE-ESY5. ライフサイクル	CE-ESY6. 要件分析 [27-1-3]	CE-ESY7. 仕様定義 [27-1-3]	CE-ESY8. 構造設計 [27-1-3, 5]	CE-ESY9. テスト	CE-ESY10. プロジェクト管理	CE-ESY11. 並行設計(ハードウェア、ソフトウェア)	CE-ESY12. 実装	
		CE-ESY13. リアルタイムシステム設計 [24-1-6, 7, 8] [26-1-1, 2, 3, 4, 5, 7, 8]	CE-ESY14. 組み込みマイクロコントローラ [24-1-3, 4, 5] [26-1-6]	CE-ESY15. 組み込みプログラム [25-1-1, 4] [26-1-6] [27-1-6]	CE-ESY16. 設計手法	CE-ESY17. ツールによるサポート [24-1-3] [25-1-5]	CE-ESY18. ネットワーク型組み込みシステム	CE-ESY19. インタフェースシステムと混合信号システム	CE-ESY20. センサ技術 [27-1-2, 3, 4, 5]	CE-ESY21. デバイスドライバ [25-1-1]	CE-ESY22. メンテナンス	CE-ESY23. 専門システム	CE-ESY24. 信頼性とフォールトトレランス		

表 3 IT 知識体系に加えて学習できる実践的知識 (1/3)

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	第9回	第10回	第11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) 加盟とパザールモデル (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) プロジェクトの発進と運営 (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			
2. 法務分野に関する知識 I	(1) オープンソース・ライセンスの法的性質 (2) オープンソース・ライセンスの分類 (3) デュアルライセンス (4) 商用ライセンスとの比較	(1) コピーレフトの概念 (2) GNU GPL (GNU General Public License) (3) MPL (Mozilla Public License)	(1) GNU LGPL (Lesser General Public License) (2) MPL (Mozilla Public License)	(1) BSD License (2) Apache Software License	(1) 著作権 (2) 特許権	(1) 商標権 (2) 実用新案権 (3) 意匠権					
3. コンピュータシステムやアーキテクチャに関する知識 I	(1) コンピュータアーキテクチャとは (2) システム (3) ネットワークアーキテクチャ	(1) コンピュータハードウェアの基 本構成 (2) コンピュータハードウェアの必要要件	(1) CPU の仕組み (2) CPU 高速化の仕組み (3) オープンソース化とのかかわり	(1) ハードディスク (2) 周辺機器の種類と特性 (3) オープンソース化とのかかわり	(1) インタフェースとは何か (2) オープンソース化とのかかわり	(1) 基本ソフトウェア (2) ミドルウェア (3) 応用ソフトウェア (4) ライセンス形態による分類	(1) OS の仕組み (2) OS の管理対象 (3) OS の分類 (4) オープンソースとのかかわり	(1) ミドルウェアのカテゴリ			
5. Linux の概念や基本操作に関する知識 I	(1) オペレーティングシステム(OS) の役割 (2) UNIX の基本的な概念 (3) Linux について (4) Linux のライセンスと企業での採用 (5) 入手方法、ディストリビューション紹介	(1) ログインとログアウト (2) プロンプトとシェルの基本操作 (3) コマンドの入力方法について (コマンド+引数) (4) ファイル操作 (5) ヘルプ機能 (6) ディレクトリ操作 (7) ファイルの読み書き (8) リダイレクトとパイプ	(1) ユーザとグループの概念 (2) ユーザの切り替え (su) (3) ユーザ/グループ操作 (4) ファイルのアクセス権操作 (5) ユーザ管理ファイル	(1) システムの現状分析 (2) サービスの管理 (3) ソフトウェアの組み込み							
6. Linux カーネルに関する知識 I	(1) OS の歴史と Linux (2) OS の概念 (3) Linux OS の基本構造 (4) カーネルの基本機能 (5) カーネルの起動プロセス (6) 参考	(1) マルチタスク キング (2) プロセス (3) コンテキストスイッチング (4) プロセススケジューラ (5) マルチプロセスへの対応 (6) スケジューリングのアルゴリズム (7) 基礎理論	(1) 割り込み (2) タイマー	(1) システムコール (2) 同期と排他	(1) プロセス管理 (2) シグナル (3) プロセス間通信						
7. Linux のシステム管理に関する知識 I	(1) システム管理業務の概要 (2) システム管理者の作業	(1) 管理者権限とは (2) Linux のインストール (3) システムの起動とサービス制御 (4) RPM パッケージ管理	(1) ファイルシステム管理 (2) ディスクの利用	(1) ユーザ管理 (2) ユーザごとのセキュリティパーミッションの	(1) バックアップ (2) ログ管理	(1) システム運用 (2) カーネルと は (2) カーネルの運用管理	(1) Linux と TCP/IP プロトコ				
8. Linux のシステムプログラムに関する知識 I	(1) ログイン手順 (2) エディタ (3) コンパイル手順	(1) シェル (2) シェル構文	(1) 低水準ファイル処理に関する説明事項 (2) ファイル操作に関する説明事項 (3) 標準入出力ライブラリに関する説明事項 (4) 書式付き入出力 (5) 一時ファイルの作成	(1) ディレクトリの保守 (2) ファイルシステムの構造 (3) /proc ファイルシステムに保持される情報	(1) コマンドラインオプションと環境変数 (2) 代表的な環境変数 (3) 共有ライブラリの利用とメモリ配置の確認	(1) static ライブラリ (2) shared ライブラリ (3) ライブラリ関数とシステムコールの違い	(1) メモリの管理 (2) ファイルのロック (3) データベース				
9. ネットワークサーバ管理に関する知識 I	(1) クライアントサーバ通信の仕組み (2) サーバの識別子 (3) JMX の概要	(1) サーバ構築に必要な環境と作業 (2) サーバ導入 (3) インストール後の設定作業	(1) ネームサーバの特徴 (2) DNS サーバ構築	(1) Web サーバ導入の内容と手順 (2) Web サーバの仕組みと作業手順 (3) セキュリティ後の設定作業	(1) メールサーバの仕組みと構成 (2) メールサーバの仕組みと作業手順						

表 3 IT 知識体系に加えて学習できる実践的知識 (2/3)

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	第9回	第10回	第11回
10. クラスタシステム構築に関する知識 I	(1) HA Cluster と HPC Cluster (2) HA クラスタ概論	(1) LVS で実現するロードバランサー (2) HA クラスタ	(1) ロードバランサーの冗長化 (2) keepalived の運用のテクニック	(1) 科学技術研究基盤としてのコンピュータシミュレーション (2) スーパーコンピュータ動向	(1) 計算の高速化と並列処理の必要性 (2) 並列計算機の種類 (3) 並列計算機の構成方式 (4) プロセッサ間のネットワーク (5) 並列処理のプログラミング (6) 並列処理の効率 (7) 数値計算の並列化 (8) 並列プログラミング環境とツール	(1) マルチスレッドプログラミング入門 (2) マルチスレッドプログラミングの基礎					
11. ネットワークアーキテクチャに関する知識 I	(1) ネットワークの構成と機能 (2) OSI 7 階層モデルとは	(1) 通信の形態とプロトコル (2) 通信プロトコルの役割	(1) インターネット通信とはどのようなものか (2) インターネット通信の構成	(1) LAN ネットワークの仕組み (2) LAN ネットワークの構成	(1) 無線ネットワークの意義 (2) 無線通信プロトコル	(1) アプリケーション通信の流れ (2) TCP/IP の仕様 (3) OSI 7 階層モデルと TCP/IP の対応	(1) IP 通信の仕組み (2) IP データグラム	(1) IP アドレスの仕組み (2) IP ルーティング (3) ルーティングテーブルのサイズと CIDR		(1) TCP の通信仕様 (2) アプリケーション通信における TCP の役割と制限	
12. ネットワーク管理に関する知識 I	(1) ネットワーク運用管理の概要 (2) マルチベンダ管理システムの分散管理の方法論 (3) ネットワーク管理ツールの種類と機能 (4) ネットワーク管理の作業	(1) 構成管理 (2) 障害管理 (3) 施設・設備管理	(1) キャパシティ管理 (2) キャパシティ管理での監視方法	(1) 性能管理 (2) 性能管理での監視方法 (3) トラフィック管理技術とは (4) RMON を使った管理	(1) TCP/IP の管理作業とは (2) トラフィック管理のためのネットワーク管理コマンドの概要 (3) ネットワーク管理コマンドの実行方法	(1) ネットワーク管理コマンドの実行方法 (2) インターネットサービスのプロセス実行 (3) RPC	(1) 機器障害の原因 (2) 障害対策の内容 (3) LAN のトラブル原因	(1) SNMP の仕様 (2) MIB (3) SNMP によるネットワーク管理仕様			
13. Java に関する知識 I	(1) Java プログラムの書き方の特徴 (2) Java のオープンソースとしての特徴	(1) Java 言語の書き方の特徴 (2) 基本的なプログラム記述の例 (3) アプレット (4) 入出力 (5) 例題	(1) オブジェクト指向プログラミング機能の実装仕様 (2) オブジェクト指向を活用したプログラミング手順	(1) Java の開発手順 (2) 統合開発環境 Eclipse	(1) ネットワークプログラミングの概要 (2) ネットワーク管理コマンドの実行方法	(1) Web アプリケーション (2) Servlet (3) JSP (4) 開発プラットフォーム	(1) JDBC の概念と接続形態 (2) JDBC プログラミング (3) Prepared Statement の利用方法とそのメリット (4) トランザクション (Transaction) 処理 (5) データソース、JDBC	(1) MVC アーキテクチャとは (2) MVC アーキテクチャのメリット (3) Servlet/JSP/JavaBeans の役割 (4) システム機能/非機能要件への対応			
14. C, C++ に関する知識 I	(1) C の概要 (2) C による開発の流れ	(1) C の書き方の特徴 (2) 基本的なプログラム記述の例	(1) 配列と文字列 (2) 文字列操作関数	(1) 関数の説明 (2) ユーザー定義関数 (3) 標準関数	(1) ポインタの基礎 (2) ポインタによるプログラミング	(1) 構造体の概要 (2) 構造体を使用したプログラミング (3) 対話型入出力	(1) コンソール入出力の説明 (2) 標準入出力 (3) ファイル操作とディレトリ操作	(1) ファイル管理の説明 (2) ファイル入出力 (3) ファイル操作とディレトリ操作	(3) ファイル操作とディレトリ操作 (2) 線形リストとは (データ構造の一例) (3) その他のデータ構造の種類と適用例 (プログラム例)		
15. Light Weight Language に関する知識 I	(1) Light Weight Language の説明 (2) Light Weight Language による開発の流れ	(1) Perl の書き方の特徴 (2) 基本的なプログラム記述の例	(1) PHP の書き方の特徴 (2) 基本的なプログラム記述の例	(1) Python の書き方の特徴 (2) 基本的なプログラム記述の例	(1) Ruby の書き方の特徴 (2) 基本的なプログラム記述の例	(1) オブジェクト指向の説明 (2) オブジェクト指向プログラミングの説明 (3) ハッシュの説明 (4) 構造化クラスの説明	(1) データ構造の説明 (2) 配列の説明 (3) ハッシュの説明	(1) データ操作の説明 (2) ファイル情報の説明 (3) 文字列の説明 (4) その他データ操作のクラスの紹介	(1) ファイル管理の説明 (2) ファイル情報の説明 (3) ファイル入出力の説明 (4) ファイル/ディレトリ操作の説明	(1) Ruby/Tk による GUI アプリケーション開発 (2) Ruby-GNOME2 による GUI アプリケーション開発 (3) GUI ライブラリの特徴と比較	
16. 開発フレームワークに関する知識 I	(1) 開発フレームワークの特徴 (2) 開発フレームワークの特性 (3) どのようにフレームワークを選択し、利用するか (4) ライセンス形態	(1) 基本的なアプリケーション記述の例 (2) 開発フレームワークの特徴 (3) フレームワークによる開発のメリット (4) フレームワークによる開発のデメリット	(1) Web アプリケーションとオープンソースフレームワーク (2) Web アプリケーション構築で利用される代表的なオープンソースフレームワーク (3) それぞれの開発内容の違い	(1) Tomcat (2) JBoss	(1) Ant (2) Maven (3) Cactus (4) Log4j (5) XDoclet	(1) 要件分析の方向 (2) 実装のプロセス					
17. 開発ツールに関する知識 I	(1) ソフトウェア開発プロセスの特徴と開発環境 (2) ソフトウェア実装 (3) アプリケーション開発	(1) ソフトウェア開発環境の必要要素 (2) 統合開発環境の構築	(1) プログラムのコンパイルとリンク (2) 統合開発環境Eclipse	(1) バージョン管理ツールの機能 (2) 主なオープンソースバージョン管理ツール	(1) 基本的なデバッグの方法 (2) 複雑に絡み合うアプリケーションとミドルウェア間のデバッグ (3) データベース処理に関連するデバッグ	(1) カーネルデバッグによるカーネルやアプリケーションのデバッグ (2) ハードウェアエミュレータの必要性	(1) 欠陥追跡システムの目的 (2) 欠陥追跡システムの機能 (2) Bugzilla とは				

表 3 IT 知識体系に加えて学習できる実践的知識 (3/3)

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	第9回	第10回	第11回	
18. 統合開発環境に関する知識 I	(1) 統合開発環境によるソフトウェア開発 (2) 統合開発環境の主な機能	(1) 統合開発環境の種類 (2) 主な統合開発環境	(1) Eclipse の背景 (2) Eclipse の特徴	(1) NetBeans の背景 (2) NetBeans IDE の特徴	(1) WideStudio の背景 (2) WideStudio の特徴	(1) Eclipse のセットアップ (2) Eclipse の基本操作	(1) Eclipse の基本操作	(1) NetBeans IDE のセットアップ	(1) NetBeans IDE の基本操作	(1) WideStudio のセットアップ	(1) WideStudio の基本操作	
19. 暗号化に関する知識 I	(1) オープンソースセキュリティの全体像 (2) 暗号化の意義と課題	(1) 共通鍵暗号方式の仕組み (2) AES の概要	(1) 公開鍵暗号方式の仕組み (2) インターネットでの公開鍵方式の重要性	(1) ソフトウェア情報の暗号化 (2) ハードウェアの暗号化 (3) 通信路の暗号化	(1) 電子証明書の種類 (2) 電子証明書の仕様 (3) 証明書発行に関わる当事者と発行までの流れ (4) CA 局による電子証明書発行、暗号化通信	(1) オープンソースOS と暗号化 (2) OSS における暗号化の実装	(1) 無線LAN 暗号化プロトコルWEP の仕様 (2) WPA の仕様	(1) 認証とはされること (2) メッセージダイジェストによる認証/改ざん防止 (3) メッセージ認証で確認	(1) VPN の構成 (2) Ipsec (3) セキュアなMPLS によるIP-VPN	(1) SSH とは-2. OpenSSH (2) IKE によるIPSec の設定	(1) SSL の概要 (2) SSL の仕様 (3) SSL の安全性 (4) SSL 通信の構成	
20. ネットワークセキュリティに関する知識 I	(1) インターネットセキュリティのリスク (2) セキュリティ実装技術 (3) ネットワークセキュリティに関連する法整備 (4) 組織におけるセキュリティポリシー	(1) コンピュータウイルスの特性 (2) コンピュータウイルスへの対処 (3) 未知のコンピュータウイルスの検出技術	(1) 攻撃手段による分類 (2) 攻撃方法の段階	(1) サーバへの侵入準備 (2) セキュリティの弱点をつくる攻撃	(1) Web のセキュリティリスク (2) 攻撃の種類と特性	(1) IP アドレスのセキュリティリスク	(1) インターネットからの侵入対策 (2) ネットワークセキュリティ設計手順	(1) ファイアウォールの機能				
21. OS セキュリティに関する知識 I	(1) インターネットセキュリティの概要 (2) リスクの構成要素の識別と評価	(1) 基本的な行動要素のチェック (2) 改ざんチェック	(1) ネットワークセキュリティの基本設定 (2) 盗聴対策	(1) ファイアウォールとDMZ アーキテクチャの種類 (2) サーバファイアウォール設定のポリシー (3) iptables によるセキュリティ管理	(1) サーバとしての管理方針と実施方法 (2) サーバでのセキュリティ実施方法	(1) セキュアシェルの基礎知識 (2) SSH の応用	(1) トンネリングの仕組みと設定 (2) IPsec による暗号化通信の導入 (3) IPsec による暗号化通信ワークショップ	(1) DNS の基本動作 (2) DNS セキュリティの原則 (3) BIND におけるセキュリティ対策 (4) djbdns				
22. RDB に関する知識 I	(1) データベースとは (2) 情報システムにおける関係データベースの役割とメリット (3) データベースの仕組みと構造	(1) RDBMS とは (2) RDBMS の機能	(1) トランザクションとは (2) 排他制御とロック	(1) データベースの構成要素 (2) 参照整合性	(1) DQA (Data Oriented Approach) (2) データモデル	(1) データベースの設計手順 (2) 分析の進め方	(1) ER モデル (Entity-Relationship Model) (2) ERD を用いたデータ設計手順 (3) ERD の効果	(1) 正規化 (2) 正規化の手順				
23. RDB システム管理に関する知識 I	(1) データベース運用管理の目的 (2) データベースの運用と管理	(1) 障害回復	(1) 最適なデータベース運用設計のアプローチ (2) 運用設計の検討項目 (3) 監視計画	(1) データベースセキュリティの概要 (2) 基本的な対策 (3) システム構成の問題	(1) リカバリのためのバックアッププラン (2) リカバリ方法の決定 (3) データベース動作環境の監視							
24. 組み込みシステムに関する知識 I	(1) 組み込みコンピュータシステムとは (2) システム (3) 組み込みコンピュータシステムとOSS の関係	(1) 組み込みコンピュータハードウェアの特徴 (2) 組み込みコンピュータアーキテクチャの必要要件 (3) OSS を活用した組み込みコンピュータアーキテクチャの事例	(1) 組み込みシステムのCPU (2) OS の要件 (3) 組み込みシステムのネットワーク形態 (4) 低消費電力	(1) データベースセキュリティの概要 (2) 基本的な対策 (3) システム構成の問題	(1) リカバリのためのバックアッププラン (2) リカバリ方法の決定 (3) データベース動作環境の監視	(1) ソフトウェア処理の基本 (2) 並行処理のアーキテクチャ	(1) 組み込みカーネルとは (2) カーネル処理の概要	(1) リアルタイム処理とは (2) アプリケーションのリアルタイム設計				
25. 組み込み開発環境に関する知識 I	(1) 組み込みシステムの特徴と開発環境 (2) 組み込み Linux 実装 (3) アプリケーション開発 (4) インタフェース利用プログラミング	(1) 組み込み開発環境の必要要素 (2) 統合開発環境の構築 (3) アプリケーション開発ツール	(1) 組み込み型コンピュータ応用システムの開発技術 (2) 組み込みコンピュータ応用システム開発支援ツール	(1) 基本的なバグの方法 (2) トレース機能 (3) 複雑に絡み合うアプリケーションとデバイスドライバ間のデバッグ (4) ハードウェアの関連するデバッグ	(1) デバッグソフトによるカーネルやアプリケーションのデバッグ (2) エミュレータの必要性	(1) ハードウェアエミュレーションとは (2) IOE とは (3) IOE によるハードウェア関連のデバッグ (4) Linux 対応 IOE	(1) ツールチェーンとは (2) ツールチェーンの種類と特徴 (3) 割り込み発生時のトラブル解消					
26. 組み込みアプリケーション開発に関する知識 I	(1) コンピュータアーキテクチャ (2) カーネルコンテキスト	(1) 非同期と同期の設計 (2) タスクの構成資源 (3) 実行環境	(1) タスクの優先順位 (2) 資源の管理 (3) カーネルによる時間管理	(1) 資源配分技術 (2) カーネルの提供する手段を使う方法 (3) デッドロック (4) 割り込みスケジューリング (ディスパッチ)	(1) 共有エリア・共有ファイル (2) カーネルによるサービス (3) デッドロック (4) 割り込みスケジューリング (ディスパッチ)	(1) MPU (2) メモリ資源の管理 (3) メモリ共有制御 (4) プログラムパーティション (ディスパッチ)	(1) プログラム資源の管理 (2) 複数タスクの同時使用 (3) 処理の優先順位					
27. 組み込みシステム最適化に関する知識 I	(1) マルチプロセッサの分類 (2) マルチプロセッサシステムの活用方法 (3) ハード面の考慮点 (4) マルチプロセッサシステムの構成例	(1) 性能の最適化 (2) CPU 高速化の仕組み (3) 入出力方式 (4) 低電力処理	(1) リアルタイムシステム設計 (2) アプリケーションタスクの設計 (3) デバイスプログラミング	(1) カーネル処理とドライバ (2) タスク設計 (3) 入出力単位 (4) タスク関連設計による最適化のアプローチ	(1) 評価項目と留意点 (2) タスク分割の設計評価 (3) MPU の評価指標ごとの評価方法	(1) MPU の評価指標ごとの評価方法 (2) プログラムと処理 (3) 最適化コンパイラ						

5.2 OSS モデルカリキュラムを習得する上で必要な前提知識

表 2 に示した IT 知識体系全体像において、網掛けを除く科目（すなわち OSS 基本知識でカバーされない知識）から、OSS 基本知識を学ぶ上で必要な前提知識を抽出したものを表 4 に示す。

ここで、「OSS 習得の前提知識」は、各 IT 知識体系の科目内容から、OSS 基本知識を習得する上で前提知識として備えておく必要がある内容を抽出したものである。また、これらの前提知識がどの OSS 科目の習得において必要になるかを表の右部分に対応表として示した。表中 OSS 基本科目の番号はそれぞれ以下の科目を表している。

1. OSSの概要に関する知識 I
2. 法務分野に関する知識 I
3. コンピューターシステムやアーキテクチャに関する知識 I (4.分散アーキテクチャは応用知識のみで構成されている)
5. Linuxの概念や基本操作に関する知識 I
6. Linuxのカーネルに関する知識 I
7. Linuxのシステム管理に関する知識 I
8. Linuxのシステムプログラミングに関する知識 I
9. ネットワークサーバ管理に関する知識 I
10. クラスタシステム構築に関する知識 I
11. ネットワーク・アーキテクチャに関する知識 I
12. ネットワーク管理に関する知識
13. Javaに関する知識 I
14. C、C++に関する知識 I
15. Light Weight Languageに関する知識
16. 開発フレームワークに関する知識 I
17. 開発ツールに関する知識 I
18. 統合開発環境に関する知識 I
19. 暗号化に関する知識 I
20. ネットワークセキュリティに関する知識 I
21. OSセキュリティに関する知識
22. RDBに関する知識 I
23. RDBシステム管理に関する知識 I
24. 組み込みシステムに関する知識 I
25. 組み込み開発環境に関する知識 I
26. 組み込みアプリケーション開発に関する知識 I
27. 組み込みシステム最適化に関する知識 I

表 4 OSS 基本知識を習得する前提となる IT 知識体系科目

分野	IT知識体系の科目名	OSS基本知識学習の前提知識	関連するOSS基本科目																										
			1	2	3	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	
組織関連事項と情報システム	IT-SP2. コンピュータの歴史	コンピュータハードウェア、ソフトウェアの歴史、インターネットの歴史、通信	○																										
	IT-ITF3. IT の歴史	コンピューティング技術の歴史、社会への影響、インターネットの歴史	○			○																							
	IT-SP4. チームワーク	共同作業、グループダイナミクス	○																										
	IT-SP7. 組織の中のIT	ビジネスプロセス、組織文化		○																									
	IT-SP9. プライバシーと個人の自由	個人情報保護法		○																	○								
ソフトウェアの方法と技術	IT-PF4. アルゴリズムと問題解決	問題解決戦略、アルゴリズムの概念、デバッグ戦略												○	○	○		○									○		
	IT-PF5. イベント駆動プログラミング	イベント処理、例外処理												○	○	○													
	IT-PF6. 再帰	再帰の概念												○	○	○													
	IT-IPT2. データ割り当てと交換	XML, DTD, XMLスキーマ, XSL, XSLT, XPath																			○								
	CE-SWE4 ソフトウェアのテストと検証	ソフトウェアテストの手法																		○	○	○							
	CE-SWE5 ソフトウェアの保守	ソフトウェアの保守、影響分析、テスト																				○							
	CE-SWE7 ソフトウェアプロジェクト管理	プロジェクト計画、ソフトウェア構成管理																				○							
	IT-SIA1. 要求仕様	要求モデリング																		○	○								
	IT-SIA5. テストと品質保証(QA)	性能評価、ユーザビリティ													○														
	IT-SIA7. アーキテクチャ	エンタープライズアーキテクチャ(EA)、システムアーキテクチャ																		○		○							
CE-CA08 コンピュータによる計算	算術演算における範囲や精度																		○	○	○								
IT-ITF6. IT 分野における数学と統計学の活用	プログラミングで利用される基本的な論理																			○	○	○		○	○				
システム基盤	IT-PT5. ファームウェア	ファームウェア、ソフトウェア、ハードウェアの概念と関連			○						○															○			
	IT-PT6. ハードウェア	デジタル論理とデジタルシステム、エレクトロニクスの基本			○																					○	○	○	
	CE-OPS5 ファイルシステム	ファイル、データの概念、ファイルシステム				○	○	○	○														○	○	○				
	CE-OPS7 OSの概要	OSの概念、歴史、基本的な機能、クライアントサーバモデル	○			○	○			○												○	○						
コンピュータハードウェアとアーキテクチャ	CE-CA02 メモリシステムの構成とアーキテクチャ	メモリシステムの階層、キャッシュメモリ			○			○	○																	○	○		
	CE-CA04 デバイサブシステム	外部記憶システム			○		○																						

6. OSS 基本知識として習得すべきポイント

OSS 基本知識の各科目毎に、学習を終了した時点で確実に習得していることが期待される概念または手法（以下「習得ポイント」）を 10 項目整理し、その内容を記す。

1. OSSの概要に関する知識 I
2. 法務分野に関する知識 I
3. コンピュータシステムやアーキテクチャに関する知識 I (4.分散アーキテクチャは応用知識のみで構成されている)
5. Linuxの概念や基本操作に関する知識 I
6. Linuxのカーネルに関する知識 I
7. Linuxのシステム管理に関する知識 I
8. Linuxのシステムプログラミングに関する知識 I
9. ネットワークサーバ管理に関する知識 I
10. クラスタシステム構築に関する知識 I
11. ネットワーク・アーキテクチャに関する知識 I
12. ネットワーク管理に関する知識
13. Javaに関する知識 I
14. C、C++に関する知識 I
15. Light Weight Languageに関する知識
16. 開発フレームワークに関する知識 I
17. 開発ツールに関する知識 I
18. 統合開発環境に関する知識 I
19. 暗号化に関する知識 I
20. ネットワークセキュリティに関する知識 I
21. OSセキュリティに関する知識
22. RDBIに関する知識 I
23. RDBシステム管理に関する知識 I
24. 組み込みシステムに関する知識 I
25. 組み込み開発環境に関する知識 I
26. 組み込みアプリケーション開発に関する知識 I
27. 組み込みシステム最適化に関する知識 I

I-1. OSS の概要に関する知識 I

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。

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

I-2. 法務分野に関する知識 I

オープンソースにまつわる知財や法務に関する知識を説明する。ライセンスの意味を説明し、具体的なライセンスを示すことで法務分野に関する理解を深めさせる。また、著作権や特許をはじめとしたソフトウェア産業に関わりの深い各種の知的財産権について解説する。

習得ポイント	説明	シラバスの対応コマ
I-2-1. OSSライセンスの意味	OSSのライセンス(使用許諾契約)に関する法的な位置づけについて説明する。とくに契約に関する有効性や、提供者、利用者、配布者といったそれぞれの関係者に対してライセンスがどのような意味を持つかについて解説する。	1
I-2-2. 目的や状況に応じたOSSライセンスの選択	代表的OSSライセンス(GPL、BSD、MPL)の特徴を示し、オープンソースソフトウェアのライセンスと商用ライセンスの違いを解説する。さらに複数のライセンスで提供しているOSSに触れ、目的に応じたライセンスの選択についても説明する。	1
I-2-3. 著作権とコピーレフト	ソフトウェア著作権とライセンスとの関係について説明する。またコピーレフトの登場した背景を説明し、コピーレフトの基本的な考え方とその具現化であるGPLとの関係について触れる。	2
I-2-4. GPLの特徴、GPL v3改訂の意義	GPL (GNU General Public License)の定義および歴史について解説する。さらに2007年に改訂されたGPL v3のポイントと意義について説明する。	2
I-2-5. GPLを取り巻く様々な問題	GPLは「GPL汚染」と呼ばれるほど波及力が強く、他のライセンスのソフトウェアとの共存が難しいことを示し、実際に発生したGPL違反の例を紹介する。	2
I-2-6. MPLの特徴、MPLとソフトウェア特許	MPL (Mozilla Public License)の特徴を解説する。OSSの特長であるバザール方式を促進しつつ、商用ソフトとの共存に配慮したこと、特にMPLでは特許との関係性が強調されていることを示す。	3
I-2-7. BSDとApacheライセンスのメリット・デメリット	BSD Licenseと、Apache Software Licenseの特徴と、最も制約が少ないゆえのメリット・デメリットを解説する。BSDライセンスの有名な「宣伝条項」について触れ、修正BSDライセンスを紹介する。	4
I-2-8. ソフトウェアの著作権	知的財産権のうち著作権の根本的な考え方について解説し、ソフトウェア産業との関わりについて基本的な概念を説明する。著作権の定義と内容を説明し、各国の制度および現状について触れる。	5
I-2-9. ソフトウェア特許をめぐる問題と動向	知的財産権のうち特許権の根本的な考えについて解説を加える。特許の定義、各国の制度および状況について触れ、さらにソフトウェア特許に対しては反対意見も多いことや、ソフトウェア特許の問題点等についても紹介する。	5
I-2-10. ソフトウェアに関連が深いその他の知的財産権	知的財産権のうち、著作権および特許以外にソフトウェア産業と関わりの深い、商標権、実用新案権、意匠権について解説する。これらの定義や意匠権を侵害した事例など、これらの権利にまつわるトピックを紹介する。	6

I-3. コンピュータシステムやアーキテクチャに関する知識 I

OSSが動作する基盤となるコンピュータシステムやアーキテクチャについて、ハードウェア、ソフトウェア、ミドルウェア等それぞれに関するアーキテクチャの種類、基本的な構成、特徴、役割について解説する。

習得ポイント	説明	シラバスの対応コマ
I-3-1. システムの構成とは	システムとは何か、コンピュータシステムという考え方について基本的な説明を行う。様々なシステムアーキテクチャが存在することを示し、システムアーキテクチャとは何か、システムアーキテクチャ設計の概要についての解説も行う。	1
I-3-2. コンピュータシステムの構成要素	コンピュータシステムの基本構成、構成する各 부품の役割、特徴、発展の歴史などについて説明する。さらにこれらの事項とOSSのかかわりに関する話題(デバイスドライバのOSS対応、オープンBIOS、ハードウェア技術情報の開示とOSS利用、等)を紹介する。	2
I-3-3. CPUの基本構成と役割	CPUの基本構成と役割、特徴、発展の歴史について解説する。代表的なアーキテクチャとしてRISCやCISCがあることを紹介し、高速化の仕組みとして各種のパイプライン処理やマルチプロセッサ構成についても触れる。	3
I-3-4. メモリの役割(メモリスステムの構成と記憶階層の概念)	メモリの基本構成と役割、特徴、発展の歴史について解説する。主記憶装置として利用されるメモリや、1次、2次キャッシュといった階層的記憶装置の概念、CPUとの接続方法などを説明し、コンピュータにおける各メモリの役割を紹介する。	2, 4
I-3-5. 周辺機器の基本構成と役割	各種の周辺機器について、その基本構成と役割、特徴、発展の歴史について解説する。周辺機器としては、補助記憶装置としてFD、MO、CD-Rなど、入力装置としてキーボード、マウス、スキャナとタブレット、出力装置としてディスプレイとプリンタを挙げる。	4
I-3-6. 様々なインタフェースの特徴	インタフェースの基本構成と役割、特徴、発展の歴史について解説する。USB、IEEE1394、eSATAなどホットプラグのインタフェースや、SCSI、IDE、PCI、シリアルといったレガシーインタフェースなど、代表的なインタフェースを個別に取り上げる。	5
I-3-7. コンピュータシステムの評価指標	コンピュータハードウェアの評価指標として、接続性(プラグアンドプレイやデバイスドライバ)、信頼性(故障率、MTTRとMTBF)、効率性(パフォーマンスとスループット、性能評価)、移植性(インタフェースの標準化、等)があることを説明する。	2
I-3-8. コンピュータソフトウェアの構成要素	基本ソフトウェア(OS)、ミドルウェア、アプリケーションといった個別のソフトウェアについて、その機能と役割、構成、発展の歴史、OSSの実装状況について解説する。またライセンス形態による分類についても触れる。	6
I-3-9. OSの基本的な構成と役割	OSの基本構成と役割、特徴、発展の歴史、OSSの実装状況について解説する。OSの仕組みについて概説し、ファイルやネットワーク、入出力インタフェースといった資源管理について説明する。	7
I-3-10. ミドルウェアの種類と役割、その特徴	Webサーバ、アプリケーションサーバ、データベースといったミドルウェアの基本構成と役割、特徴、OSSの実装状況について解説する。また、それぞれに関して代表的なOSSを紹介する。	8

I-4. 分散アーキテクチャに関する知識

(レベルⅡのみから構成されるため本報告書では割愛)

I-5. Linux の概念や基本操作に関する知識 I

Linux を利用するにあたり必要となる UNIX の知識や、Linux 発展の歴史、開発モデル、ディストリビューションの特性といった話題を紹介する。さらにシェルを介した OS の操作、マルチタスク・マルチユーザの概念、システム管理方法などについて概説する。

習得ポイント	説明	シラバスの対応コマ
I-5-1. UNIXとLinuxの関係、Linux開発の歴史	オペレーティングシステム(OS)の役割と、Linuxの正本となったUNIXの基本的な概念を説明する。さらにUNIXとLinuxの関係、Linux開発の歴史について述べる。	1
I-5-2. Linuxの開発モデルと利用上の注意点	Linux開発モデルの特徴であるバザール形式による開発スタイルを紹介する。また、LinuxおよびLinuxで動作するアプリケーションの利用に関するポイントとして、ライセンスに関する事項、ビジネス事例、典型的な利用方法などを解説する。	1
I-5-3. Linuxの配布形態	Linuxディストリビューションとは何か、その定義と特徴について説明し、代表的なディストリビューションを挙げる。また各ディストリビューションの特徴や系統(有償版/無償版、サーバ向け/デスクトップ向け)を説明する。	1
I-5-4. ユーザの概念とログイン・ログアウト	ユーザの概念について触れ、ログイン、ログアウトの手順、およびBシェル(bash)を利用したUNIX/Linuxの基本操作を説明する。またシェルを介したOSとの対話の基本的な考え方について解説する。	4,5
I-5-5. ファイルシステムの木構造とその操作	ファイルシステムは木構造をしていること、各種の種類があることなど、ファイルシステムの特徴について解説する。基本的なファイル操作/ディレクトリ操作のための代表的なコマンドとその使い方について説明する。	2,3
I-5-6. オンラインマニュアルの利用方法	オンラインマニュアルとして、manコマンドやinfoコマンドの概要を説明する。また具体的な利用方法やどのような情報を得ることができるかについて解説する。	6,7,8
I-5-7. ファイルの読み書きと入出力ストリーム	ファイルを読むためのコマンドとしてcatおよびページャ(more, less)を紹介する。さらに入出力ストリームの概念に触れ、リダイレクトやパイプの概念について解説する。	6,7,8
I-5-8. エディタの利用方法	エディタを用いてテキストファイルを編集するという概念について解説する。また、viやemacsなど一般によく利用されているエディタの基本的な操作方法を説明する。	6,7,8
I-5-9. 複数の仕事・アプリを並行処理するマルチタスク	UNIX/Linuxで処理を行う際の基本となるマルチタスクの概念を説明する。シングルタスクとマルチタスクの違い、さらにシングルユーザとマルチユーザの違いについて触れ、タイムシェアリングでマルチタスクを実現していることを説明する。	4,5
I-5-10. マルチユーザとグループ、ユーザ権限	UNIX/Linuxにおけるユーザの概念を説明する。マルチユーザ環境の意義と利点を示し、その必要性について説明する。さらに各ユーザの権限について解説し、ユーザの役割、グループによる管理方法などについて触れる。	4,5

I-6. Linux カーネルに関する知識 I

OSS による OS の代表的な存在である Linux オペレーティングシステムを題材として、その内部構造の基本と、動作原理の基本的な概念を解説する。

習得ポイント	説明	シラバスの対応コマ
I-6-1. 代表的なオペレーティングシステム	OSの基本的な役割を説明し、OSがコンピュータに搭載されるようになった経緯と現在に至る発展の歴史を解説する。さらに現在の代表的なOSを紹介し、それぞれのOSが持つ特徴や各OSの分類についても触れる。	1
I-6-2. OSの基本構造とカーネルの役割	Linux OSの基本的な構造を説明する。また中心的な役割を担うカーネルの基本的な機能として、プロセス管理、メモリ管理、ファイル管理、ネットワーク、I/Oなどの概要を紹介する。	2,5
I-6-3. マルチタスクOSにおけるプロセスの概念	「プロセス」の概念と、プロセスを切り替えて時分割でタスクを遂行するマルチタスクの概念について解説する。またOSが各プロセスを管理する手法についても言及する。	5
I-6-4. プロセスの切り替えとスケジューリングアルゴリズム	コンテキストスイッチの概念について触れ、プロセスが切り替わるメカニズムの概要を説明する。スケジューリングアルゴリズムの実際を紹介し、さらにその基礎となる理論である待ち行列議論とマルコフ連鎖について解説する。	2,5
I-6-5. 割り込みと時分割処理	CPUとその他デバイスの関係について整理し、「割り込み」の概念を説明する。割り込みの種類や割り込みの管理など割り込みに関連するいくつかの話題について触れ、さらに、カーネルにおける割り込み処理の概要を解説する。また時分割処理に関わる基本的な事項を説明する。	3
I-6-6. システムコール	一般のアプリケーションからOSの機能を利用する際に使用するシステムコールを紹介する。システムコールの位置づけを説明し、システムコールが呼ばれたときのOS内部での振る舞いについて概説する。	3,4
I-6-7. プログラムとOSの動作モード	プログラムが動作するプロセス空間という概念を説明し、その目的や特徴、それぞれの空間でできることとできないことを示す。さらにシステムコールによる動作モードの遷移について解説する。	4,5
I-6-8. 同期と排他制御	プロセスの同期や排他が必要になる状況(デッドロック等)やカーネルによる管理について述べ、排他制御を実現するための実装技術やその背景となる基礎理論について触れる。またカーネルの内部で排他制御が実装されている具体的な例を示して説明する。	4,5
I-6-9. プロセスのライフサイクル	プロセスが生成されてから終了して消滅するまでのライフサイクルについて解説する。プロセスの内部データ構造、状態遷移、プロセスグループ、プロセスを生成する方法など、プロセスのライフサイクルに関連する一連のトピックを紹介する。	2,5
I-6-10. プロセス間通信	プロセスの間でデータやメッセージを通信するための手段として提供されている共有メモリ、セマフォ、メッセージなどを紹介する。	4,5

I-7. Linux のシステム管理に関する知識 I

Linux のシステム管理に関する基本的な手順を解説する。システム管理者の役割を示し、システムのインストールから各種サービスの設定、システムの起動・停止とシステムの運用など、実際に Linux システムを運用するために必要なノウハウを具体的に説明する。

習得ポイント	説明	シラバスの対応コマ
I-7-1. システム管理者の役割と管理業務	システム管理業務の概要と、目的、必要性、システム管理作業の種類と内容などについて解説する。またシステム管理者の役割と管理者権限について説明し、システム管理者として心がけるべき事項についても言及する。	1,2
I-7-2. Linuxシステムのインストール	Linuxのインストール作業について解説する。インストールの具体的な手順を示し、各種項目の設定方法やアプリケーションパッケージの導入方法、パッケージ管理、インストール作業を行う際に気をつけることなどについて説明する。	3
I-7-3. Linuxシステムの起動と停止	Linuxシステムの起動と終了の手順、起動時および終了時の動作内容を説明する。またサービスの起動と停止の基本について述べ、提供するサービスの選択方法やサービスの起動停止を設定する各種のツールについて解説を加える。	2,3
I-7-4. 各種サービスの設定方法	Linuxシステムが提供する各種のサービスについて、多くのサービスでは/etc 以下に設定ファイルが集められていることを示し、さらに代表的なサービスの設定方法や設定ツールについて解説する。	2
I-7-5. ファイルシステムとディスクの管理	Linuxにおけるファイルシステムについて説明し、ファイルシステムを管理する方法を紹介する。また、ハードディスク、フロッピーディスク、CD、DVD、その他の周辺機器など様々な形態のディスク装置について、その利用方法を解説する。	6
I-7-6. ユーザの登録・削除とユーザ環境の整備	Linuxにおけるユーザの登録・削除等のユーザ管理について、その基本と作業手順、ユーザ環境の設定方法、グループの作成方針、ユーザごとのセキュリティ管理など、実際の作業に必要な項目を解説する。	4
I-7-7 システムのバックアップとリストア	Linuxシステムにおけるデータやアプリケーションリソースに関して、不慮の事故に備えたバックアップの運用管理方法について説明する。さらに事故が生じた場合のリストア方法や運用コストを考慮したバックアップ方針についても説明する。	5
I-7-8. ログの取得、管理と解析	システムが正常に動作しているかどうかを監視する手段としてのログ管理について解説する。ログの種類、ログの取得方法、ログの分散管理といった話題や、ログ取得のタイミング、ログの解析方法といったログ運用にまつわる話題についても触れる。	5
I-7-9. カーネルの運用・管理とカーネル再構築	カーネルの位置づけと機能についてカーネル管理の側面から説明し、カーネルの設定を管理する運用の重要性を示す。また実際にカーネルを更新する方法やカーネルパラメータの調整、カーネルモジュールの取扱い方法などについて説明する。	7
I-7-10. ネットワークの基本的な設定	Linuxにおけるネットワークの管理運用方法について解説する。イーサネットやWifiといった各種のネットワークの設定方法や設定ファイル、稼働状況の確認方法、ネットワーク全体の設定ファイルなど、ネットワーク運用に関わる基本的な項目を確認する。	8

I-8. Linux のシステムプログラミングに関する知識 I

Linux においてシステムが提供するリソースを活用してプログラムを実行するシステムプログラミングについて解説する。プログラムの作成方法から始まり、プログラムからのファイル操作、パラメータの受け渡し、ライブラリの利用、メモリ管理やデータベースの利用までを説明する。

習得ポイント	説明	シラバスの対応コマ
I-8-1. エディタを用いたソースファイルの作成	Linuxのシステムへログインし、エディタを起動してソースファイルを作成するまでの手順を説明する。また代表的なエディタであるviとemacsについて、基本的な操作方法を解説する。	1,4
I-8-2. コンパイラの仕組みと実行ファイルの作成	C言語プログラムを題材として、コンパイラの仕組み、プリプロセッサ、コンパイラ、リンカの役割を解説する。またソースプログラムからオブジェクトファイルを生成し、実行プログラムを構築するまでの手順やコンパイラオプションの指定などを説明する。	1
I-8-3. シェルスクリプトによる定形処理	シェルの種類やシェルスクリプトの概念、対話的シェルの操作方法を説明する。スクリプトを作成してから実行するまでの手順を示し、また基本的なシェルコマンドやシェルスクリプトの文法、記述方法について言及する。	1,2
I-8-4. 低水準ファイル処理によるファイル操作	低水準ファイルアクセスを行うシステムコールを使ってファイルを操作する方法を示す。ファイル操作のモード、ファイルの生成、オープンとクローズ、ファイルの読み書きとファイルパーミッションの変更、その他ファイル操作に必要なシステムコール群について説明する。	3
I-8-5. 標準入出力ライブラリによるファイル操作	標準入出力ライブラリを用いてファイルを操作する方法を示す。低水準ファイル処理のそれぞれに対応する各関数を紹介し、さらに書式付き入出力や利用上の留意点も示す。また一時ファイルを作成する関数についても触れる。	3
I-8-6. ファイルシステムの操作	ファイルシステムの概念や構造、ファイルシステムの種類と各ファイルシステムの特徴について説明し、複数のディスクブロックをひとつのツリーとして扱う考え方とその方法を解説する。またディレクトリをプログラムから操作する方法や特殊なディレクトリについて説明する。	4
I-8-7. コマンドラインオプションと環境変数のプログラムへの渡し方	シェルからプログラムへパラメータを渡す方法として、コマンドライン引数による方法と環境変数を用いる方法を説明する。渡されたパラメータをC言語プログラム内部で処理する方法やプログラムからの返り値の取扱いについて解説し、代表的な環境変数についても説明する。	5
I-8-8. ライブラリとプログラムの関係	プログラムをコンパイルしてできあがる実行ファイルと、実行ファイルから関数呼び出しで利用するライブラリとの関係を説明する。また共有ライブラリの考え方や、実行時の動作、lddコマンドによる利用ライブラリの一覧やsizeコマンドによるメモリ利用状況の把握などを解説する。	5
I-8-9. ライブラリの種類とライブラリの実用方法	ライブラリにはstaticライブラリとsharedライブラリの2種類があることを説明し、それぞれの利点と特徴、利用方法について解説する。また、ライブラリを利用したシステムプログラミングの例を示す。ライブラリで提供される関数とシステムコールの違いについての説明も行う。	6
I-8-10. メモリ管理、ファイル管理とデータベースの利用	OSが管理するメモリの動的な利用方法、ファイルの排他処理として行うロックの方法、Berkeleyデータベースの利用など、システムプログラムから大規模なデータを管理する方法について解説する。	7

I-9. ネットワークサーバ管理に関する知識 I

OSS が動作するネットワークサーバについて、その基本的な役割と構成、種類、特徴を解説する。さらに代表的なネットワークサーバを紹介し、各サーバの仕組みや実装について説明する。

習得ポイント	説明	シラバスの対応コマ
I-9-1. サーバの役割と通信プロトコル	ネットワークにおいてサーバが担う役割やその基本的な仕組み、ネットワークサーバの機能、特徴を説明する。さらに通信プロトコルについて言及し、クライアントとサーバの通信によるサービスの提供方法について説明する。	1
I-9-2. ネットワークアドレス、ドメイン名やホスト名の意味	インターネットを構成する各ノードを識別する方式として、ネットワークのアドレス(IPアドレス)とドメイン名、ホスト名の構成を概説する。さらにネットワークバイトオーダーやURI (Uniform Resource Identifier)など、関連するトピックを紹介する。	1
I-9-3. サーバの導入と設定方法	一般的なネットワークサーバ導入作業の概要と手順を、OSや必要なソフトウェアパッケージの入手、インストール、起動環境設定、ネットワーク設定、サービス提供内容の設定、サーバシステムの起動と停止など、順をおって具体的に示す。	2
I-9-4. 名前解決の仕組みとDNS	DNS (Domain Name System)の概要を説明し、DNSサービスを提供するDNSサーバの仕組みとDNSプロトコルについて解説する。またOSSによるDNSサーバ実装の歴史と背景など関連する情報も紹介する。	3
I-9-5. DNSサーバの構築・設定方法	Linuxで動作するネームサーバを導入、構築し、設定を行う手順を解説する。設定ファイルの各項目について、その意味と書き方を紹介し、実際に設定ファイルを作成する手順を示す。またネームサーバが実際に動作する状況について解説する。	3
I-9-6. Webサーバの仕組み	WWWの発展とWebサーバの機能や役割、CGIによるアプリケーション実行や拡張について説明する。OSSによるWebサーバ実装の歴史と背景、代表的なサーバの特徴についても述べる。また、HTTP (HyperText Transport Protocol)の概要と通信方式を解説する。	4
I-9-7. Webサーバの構築・設定方法	Linuxで動作するWebサーバを導入、構築し、設定を行う手順を解説する。設定ファイルの各項目について、その意味と書き方を紹介し、実際に設定ファイルを作成する手順を示す。またWebサーバが実際に動作する状況について解説する。	4
I-9-8. メールサーバの仕組み	ネットワークでメールをやりとりするために用意されるメールサーバの基本的な構成を解説する。メールを送信するSMTP (Simple Message Transfer Protocol)、受信したメールをクライアントにダウンロードするPOP (Post Office Protocol)など、関連するプロトコルを紹介する。	5
I-9-9. SMTPサーバの構築・設定方法	Linuxで動作するSMTPサーバを導入、構築し、設定を行う手順を解説する。また設定ファイルを作成する手順を示す。またメールサーバが実際に動作する状況について解説する。	5
I-9-10. POPサーバの構築・設定方法	Linuxで動作するPOPサーバを導入、構築し、設定を行う手順を解説する。また設定ファイルを作成する手順を示す。またメールサーバが実際に動作する状況について解説する。	5

I-10. クラスタシステム構築に関する知識 I

HA (High Availability) クラスタと HPC (High Performance Computing) クラスタについて説明し、クラスタシステム構築に必要な技術を解説する。またクラスタシステムにおけるプログラミング技術として必須である並列プログラミングやマルチスレッドプログラミングの基礎も紹介する。

習得ポイント	説明	シラバスの対応コマ
I-10-1. クラスタシステムの種類(HAクラスタとHPCクラスタ)	高可用性(HA)クラスタと高性能(HPC)クラスタについて、それぞれの概要と特徴、背景と歴史を説明する。また、各クラスタシステムの目的と位置づけを比較し、各システムに固有の技術を概説する。	1
I-10-2. 負荷分散(ロードバランサー)の種類と構成	HAクラスタの基本技術である負荷分散技術について、その基本的な概念、種類と構成および構成要素を解説する。負荷分散を実現するロードバランサーの様々な実装(レイヤ4/7)を紹介し、それぞれの構成、特徴について説明する。	1
I-10-3. 負荷分散アルゴリズムと死活監視	負荷分散を実現する代表的なアルゴリズム(ラウンドロビン等)を紹介し、その具体的な手順を説明する。またサーバの死活監視技術について触れ、HAクラスタ運用時に発生する作業について説明する。	1
I-10-4. LVS (Linux Virtual Server)によるロードバランサーの実現方法	LVS (Linux Virtual Server)を利用してHAクラスタを構築する方法について解説する。その構成要素を説明し、IPVSカーネルの構築、ipvsadmおよびkeepalivedのビルド、テスト環境の構築、DSR (Direct Server Return)の構築といった具体的な手順を示す。	2
I-10-5. VRRP(Virtual Router Redundancy Protocol)による冗長化	ルータを冗長化するVRRP (Virtual Router Redundancy Protocol)とは何かについて解説し、ネットワークの構成方法からkeepalivedの設定といった冗長化の実現方法を示す。さらにkeepalived運用時の留意点など実践的なポイントを紹介する。	3
I-10-6. スーパーコンピュータの歴史とクラスタによる実現	科学技術計算基盤としてのスーパーコンピュータについて、その概要と歴史、最近の動向を紹介する。近年急増しているクラスタによるスーパーコンピュータの実現方法について触れ、PCクラスタに関するいくつかのプロジェクトを紹介する。	4
I-10-7. 並列計算機の種類と構成	並列プログラミングの概念を理解するために、まず計算の高速化と並列処理の必要性を示す。また並列計算機の種類について触れ、プロセッサやメモリの結合方式、バスのトポロジーなど、構成方式の違いについて解説する。	5
I-10-8. 並列処理プログラミングの基本、並列化処理	並列プログラミングの基本的な技法と、並列処理の効率性について説明する。代表的なアルゴリズムについて並列化の技法を紹介し、さらに並列プログラミング環境と並列プログラミングで利用できるツールを紹介する。	5
I-10-9. マルチスレッドプログラミングの基礎	並列プログラミングの一種であるマルチスレッドプログラミングの基礎について解説する。スレッドとは何か、スレッドとプロセスの違い、スレッドの生成と終了など、スレッドプログラミングに関する基本的な概念を説明する。	6,7
I-10-10. スレッド間の通信とスレッドの同期	pthreadを題材として、具体的なマルチスレッドプログラミング方法を示す。またマルチスレッドプログラミングで必須の技術である同期とスレッド間通信の実現方法を紹介する。	6,7

I-11. ネットワークアーキテクチャに関する知識 I

TCP/IP プロトコルの全体像について解説し、TCP/IP を用いた通信で構成されているインターネットや LAN (Local Area Network) の仕組みと特徴を説明する。また OSI の 7 層モデルと対比しつつ、現代におけるコンピュータネットワークの階層構造について説明する。

習得ポイント	説明	シラバスの対応コマ
I-11-1. ネットワークの構成とOSI7層モデル	IT社会の基盤となる基本的なネットワークの構成と、各構成要素の機能について解説する。またOSIの7階層モデルについて言及し、階層構造を成す意味と、物理層からアプリケーション層まで各層の機能を説明する。	1,2
I-11-2. 通信形態と通信プロトコル	通信の形態と通信プロトコルに関して、その種類、各方式の特徴、通信プロトコルの役割などを解説する。通信プロトコルについては、物理層からトランスポート層まで、各層に対応するプロトコルの種類と特徴を示す。	2
I-11-3. インターネットの構成と特徴	インターネットの通信について、その仕組み、構成要素と特徴を説明する。またインターネット通信を構成する要素として、ネットワークのハードウェアと物理構成、論理構成について解説する。	3
I-11-4. LAN(Local Area Network)の仕組みと特徴	LAN (Local Area Network)について、データリンク層におけるプロトコルの観点からその仕組みと特徴を解説する。CSMA/CDのプロトコルを説明し、コリジョンを回避する方法、スイッチング技術なども紹介する。またVLAN (Virtual LAN)の話題についても触れる。	4,6
I-11-5. 無線ネットワークの種類と特徴	無線通信によるネットワークの仕組みと特徴、有線ネットワークとの違いと留意点などを解説する。具体的には、現在ひろく利用されているIEEE802.11および802.15のプロトコルとその内容について言及する。	5
I-11-6. TCP/IPの仕様とOSI7層モデルとの対応	インターネットにおける通信の概要を説明し、インターネット上におけるアプリケーション間の通信がどのように実現されているかを示す。またTCP/IPの仕様を解説し、TCP/IPとOSIの7層モデルの対応関係を説明する。	6
I-11-7. IP通信の仕組みとデータグラム	IP通信の形態と通信プロトコルの種類、IPの基本機能や通信の仕組み、特徴について解説する。またIPパケット(IPデータグラム)の構造や取扱い方など、IP通信に関連するいくつかのトピックを紹介する。	7
I-11-8. IPアドレスとルーティングの基礎	IPアドレスの構成と、インターネット上に存在するふたつのノード間の通信を実現するために考案されたルーティングの仕組みを説明する。ルータの機能、ルーティングテーブル、デフォルトゲートウェイなど、ルーティングを実現する個々の概念について解説を加える。	8
I-11-9. ルーティングプロトコル	ルーティングプロトコルの種類とその機能について解説する。具体的には、経路制御情報を交換するために利用されるRIP (Routing Information Protocol)やRIPを改良して提案されたOSPF (Open Shortest Path First) プロトコルの仕様と特徴について述べる。	9
I-11-10. TCP通信の仕組みと応用	TCP通信の仕様、役割とその特徴について解説する。アプリケーション通信におけるTCPの役割やTCP通信の制限事項について触れる。また同じトランスポート層の通信プロトコルであるUDP (User Datagram Protocol)と比較し、その特質に関して説明する。	10

I-12. ネットワーク管理に関する知識 I

ネットワークの運用管理に関して、実際の作業に必要な知識を説明する。各種管理の作業内容、ネットワーク管理コマンドの利用方法など実際の作業手順、ネットワーク障害の発生原因やトラブルシューティングに関する知識など、実務的なノウハウを解説する。

習得ポイント	説明	シラバスの対応コマ
I-12-1. ネットワーク運用管理の概要	ネットワーク運用管理の全体像と各運用管理で行う作業の概要を説明する。インターネット環境の運用リスクや、マルチベンダ/マルチプロトコル環境に対する分散管理の方法論や、分散管理サービスの事例についても触れる。	1
I-12-2. ネットワーク運用管理に求められる作業	ネットワーク管理に求められる個々の作業について、その目的や位置づけを概説する。また各種ネットワーク管理ツールの紹介、ネットワーク管理に必要な具体的な作業について解説する。	1
I-12-3. 構成管理、障害管理、施設・設備管理の作業手順	ネットワーク管理に必要な個別の作業のうち、構成管理、障害管理、施設・設備管理を解説する。各作業の目的、内容、作業手順について述べ、実際に作業を行う際に注意すべきポイントを示す。	2
I-12-4. キャパシティ計画の立案からキャパシティ監視まで	ネットワーク管理に必要な個別の作業のうち、ネットワークサービスのリソース最適化を担い、もともと複雑で重要な作業であるキャパシティ管理について説明する。キャパシティ計画の立案、キャパシティ評価指標の決定、キャパシティ管理における監視方法などについて述べる。	3
I-12-5. 性能管理とトラフィック監視	ネットワーク管理作業において重要な作業である性能管理とトラフィック監視技術について解説する。通信のボトルネックや問題点の洗い出し方法、トラフィック監視に利用するツール、トラフィック情報の分析方法など、性能管理に関する個別のトピックを説明し、さらにRMONを使った遠隔監視方法についても触れる。	4
I-12-6. TCP/IPネットワークにおける管理作業	TCP/IPネットワークにおいて求められる具体的な管理作業について説明する。ネットワーク層におけるIPの動きを解説し、IPレベルの管理手法について解説する。	5
I-12-7. ネットワーク管理コマンドの利用方法	TCP/IPネットワークにおける、基本的かつ頻繁に利用するネットワーク管理コマンドとして、ping、arp、netstat、tracert、nslookup (dig、host)、ifconfigなどを紹介し、その利用方法を解説する。	5
I-12-8. 各種ネットワークサービスの開始	ネットワークサーバ運用管理の目的、内容、特徴などについて述べ、各種ネットワークサービスの設定方法やサービス開始の手順を解説する。また、well-knownポート、/etc/servicesによるサービスの定義など、ネットワークサービス設定に必要な知識について説明する。	6
I-12-9. ネットワーク機器の障害対策とトラブルシューティング	ルータやスイッチなど、ネットワーク機器運用管理の基本を示し、電源電圧障害、ケーブル障害、熱暴走、設定ミスなど起こりうる機器障害の内容とその対策について解説する。またハードウェアトラブルの原因を追求する手法についても述べる。	7
I-12-10. SNMPの仕組みと、SNMPによるネットワーク管理方法	ネットワーク運用管理に利用するプロトコルであるSNMP (Simple Network Management Protocol)の概要と仕組み、特徴を紹介する。SNMPで取り扱うMIB (Management Information Base)について言及し、さらにSNMPによるネットワーク管理の手法について説明する。	8

I-13. Java に関する知識 I

Web アプリケーション構築で利用されることの多い Java 言語について、基本的な仕組み、プログラミング文法、オブジェクト指向によるプログラム設計の概念、各種のアプリケーション開発を解説する。さらに Web アプリケーション開発やデータベースの利用についても述べる。

習得ポイント	説明	シラバスの対応コマ
I-13-1. Java言語の歴史、特徴、思想と背景	Javaの基本概念と特有用な機能、発展の歴史、Javaが登場した背景と必然性、Java利用のメリットなど、Javaプログラミングを理解する前におさえておくべき項目として、Java言語にまつわる様々なトピックを紹介する。	1
I-13-2. Javaを用いたプログラミング方法	Javaプログラムの例を示し、Javaプログラムの記述からコンパイル、実行までの手順を紹介する。さらに様々なJavaのエディションや実行環境の整備など、Javaを用いたプログラミングを実施する際に留意すべき点について述べる。	1
I-13-3. Java言語の基本的な構造、型、演算子、制御構文	Javaの基本的な仕組み、特徴、構造を説明する。基本的なプログラム要素として、識別子、変数、型、クラス、演算子、制御構文について解説する。またアプレットやサーブレットといった形態、クラスライブラリの利用についても言及する。	2
I-13-4. Javaによるオブジェクト指向プログラミング	Javaの大きな特徴であるオブジェクト指向プログラミングについて、その基本的な特徴とメリット、C言語との共通点や違いを説明する。クラス、オブジェクト、メソッド、コンストラクタなどオブジェクト指向の様々な概念を紹介し、オブジェクト指向による代表的なプログラミング手法を示す。	3
I-13-5. Javaによるアプリケーション開発、各種リソースの利用	Javaによるアプリケーション開発の基本的な手順を解説する。既存のドキュメントへアクセスする方法、必要なライブラリの探し方、オブジェクト指向を活用したJavaプログラミング開発技法などを紹介する。	4
I-13-6. Eclipseを用いたJavaプログラミング	統合開発環境Eclipseを用いたJavaプログラミングを概説する。Eclipseの利用方法と活用のメリット、Eclipseのプラグインによる拡張など、Eclipseを活用したJavaアプリケーション開発に役立つ様々な話題についても触れる。	4
I-13-7. Javaによるネットワークプログラミング	Javaによるネットワークプログラミングの概要、特徴を説明する。ネットワークプログラミングの基礎であるソケットを用いたセッション管理について、Javaを例題として説明する。またJavaのRMI (Remote Method Invocation) を活用した分散プログラミングも解説する。	5
I-13-8. ServletとJSPによるWebアプリケーション開発	Webアプリケーションの概要や特徴について説明し、さらにServletとJSPによるWebアプリケーション開発の手順と内容について解説する。また開発プラットフォームとしてのTomcatやJBossといった代表的なOSSの実装を紹介する。	6
I-13-9. JDBCによるデータベースアクセス方法	Webアプリケーションではほぼ必須となるデータベースとの連携を実現する方式として、JDBCを用いたJavaアプリケーションとデータベースの接続方法を説明する。その基本的な概念、プログラミング方法、トランザクション処理などについて解説する。	7
I-13-10. MVCアーキテクチャの基本と特徴	Javaアプリケーションを例題として、Webアプリケーションの基本アーキテクチャとして想定されるMVC (Model-View-Controller)モデルの内容とメリットについて解説する。それぞれを構成する要素技術を示し、モデルとの対応関係について説明する。	8

I-14. C、C++ に関する知識 I

主要なオープンソースソフトウェアの記述言語である C 言語の基本的な知識を解説する。C 言語プログラムの構造、型、演算子、制御構文や標準的な関数、ポインタや構造体、ユーザや外部データとの入出力に求められるプログラミングなどについて説明する。

習得ポイント	説明	シラバスの対応コマ
I-14-1. C言語の歴史と特徴、開発事例と開発手順	Cプログラミング学習の入り口として、C言語の歴史や特徴、他の言語との比較、Cによる開発事例を解説する。また、プログラムを開発する手順として、エディタによるプログラム作成からコンパイル、プログラム実行までの流れを説明する。	1
I-14-2. C言語の基本的な構造、型、演算子、制御構文	Cの基本的な仕組み、プログラムの構成、基本的な文法を説明する。基本的なプログラム記述の例を示し、変数、データ型、演算子、制御構文といったCプログラムの基本的な要素について説明を加える。	2
I-14-3. 関数の定義、標準関数の利用方法	関数とは何かを説明し、関数を活用したプログラム部品化の概念を示す。関数定義の方法と関数の利用方法、変数のスコープなど、関数の利用時に気をつけるべき点を列挙する。さらに代表的な標準関数を紹介し、標準で提供される関数で様々なことを実現できることを例示する。	4
I-14-4. ポインタを利用した効果的なプログラミング	メモリ上に配置されているデータの抽象化とアドレスの考え方を示し、ポインタの概念、特徴、利用方法について解説する。また配列や文字列、関数を利用する際にポインタを上手に利用する技法を示し、関数ポインタの概念についても言及する。	3, 5
I-14-5. 構造体の概念と構造体を使用したプログラミング	構造体の概要、定義と使い方を解説し、変数や配列などデータをまとめて扱う方法について説明する。また構造体を配列で利用する方法、構造体メンバへのポインタによるアクセス、関数への構造体データの受け渡し方法など、実際にプログラムで構造体を利用する際に必要となる技術を解説する。	6
I-14-6. マクロの利用 (プリプロセッサ機能)	Cコンパイラにおけるプリプロセッサの位置づけと、プリプロセッサが備えている機能を説明する。またマクロによるプログラミング例を紹介し、マクロの効果的な使い方とマクロ利用時の留意点について解説する。	2
I-14-7. 標準入出力を利用したプログラミング	Cプログラムにおける入出力の概念を説明し、入出力ストリームに対するプログラミング手法を紹介する。ここではとくに標準入出力を対象とした文字や文字列の受け渡しについて解説する。	7
I-14-8. ファイル入出力とファイル/ディレクトリ操作	ファイル入出力、ファイル操作、ディレクトリ操作といったCプログラムからファイルにアクセスするファイル管理について説明する。またファイルに対する高水準ファイル入出力関数と低水準ファイル入出力があることを示し、具体的なデータ入出力方法を解説する。	8
I-14-9. データ構造	データ構造とは何かを解説し、線形リスト、ツリー、スタック、キューといった一般的なデータ構造を紹介する。またメモリ管理とデータ構造の関係について触れ、Cプログラムによるデータ構造の実装例を紹介する。	9
I-14-10. アルゴリズム	用途に応じたデータ構造の使い方を説明し、それぞれのデータ構造を使用した代表的なアルゴリズムや効果的な利用方法を紹介する。	9

I-15. Light Weight Language に関する知識 I

軽量プログラミング言語(Light Weight Language)と呼ばれる言語について解説する。代表的な Light Weight Language である Perl、PHP、Python、Ruby を紹介し、その特徴と基本的な構文を説明する。さらに Ruby を利用したオブジェクト指向プログラミングや Ruby 特有の項目を掘り下げて解説する。

習得ポイント	説明	シラバスの対応コマ
I-15-1. Light Weight Languageの歴史や特徴、種類、使い方	Light Weight Languageの概要を紹介し、その歴史、特徴、用途、代表的なLight Weight Languageの種類などについて概説する。さらにLight Weight Languageを利用した開発の流れについて述べ、プログラミングから実行までの手順を示す。	1
I-15-2. Perlの特徴と正規表現	古くから利用されているPerlの概要と特徴について触れ、Perlの動作、ライセンス形態、代表的な利用方法などを紹介する。また基本的なプログラミングができるように、型、演算子、制御構文などについて説明し、Perlを特徴付ける「正規表現」について解説する。	2
I-15-3. PHPの特徴と埋め込みプログラミング	Webアプリケーションのロジック記述に利用されることの多いPHPの概要と特徴について触れ、PHPの動作、ライセンス形態、代表的な利用方法などを紹介する。また基本的なプログラミングができるように、型、演算子、制御構文などについて説明し、PHPを特徴付ける「埋め込みプログラミング」を解説する。	3
I-15-4. Pythonの特徴と連想配列	設定ファイルや機能拡張言語のベースとしてしばしば利用されるPythonの概要と特徴について触れ、Pythonの動作、ライセンス形態、代表的な利用方法などを紹介する。また基本的なプログラミングができるように、型、演算子、制御構文などについて説明、Pythonを特徴付ける「連想配列」を解説する。	4
I-15-5. Rubyの特徴、基本的な構造、型、演算子、制御構文	オブジェクト指向プログラミングを特徴とするRubyの概要と特徴について触れ、Rubyの動作、ライセンス形態、代表的な利用方法などを紹介する。また基本的なプログラミングができるように、変数、演算子、制御構文などについて解説する。	5
I-15-6. Rubyによるオブジェクト指向プログラミング	Rubyを題材として、オブジェクト指向プログラミングの基本について解説する。オブジェクト指向の概念を説明し、クラスやメソッドの定義、継承、カプセル化、ポリモルフィズムといった考え方がRubyでどのように表現されるかを示す。	5
I-15-7. Rubyが持つ特徴的なデータクラス(配列、ハッシュ等)	Rubyに組み込まれている配列やハッシュなどの汎用データ型クラスについて説明する。配列、ハッシュ、構造化クラスの概要を解説し、各クラスの構成や使い方を紹介する。	7
I-15-8. Rubyにおけるデータ(数値、文字列、その他)の操作	Rubyに組み込まれている数値や文字列、日付などの固有なデータ操作に特化したクラスについて説明する。数値クラス、文字列クラス、その他特有のクラスについて、その概要を解説し、各クラスの構成や使い方を紹介する。	8
I-15-9. Rubyによるファイル操作のプログラミング	Rubyに組み込まれているファイル入出力やファイル/ディレクトリ操作などのファイル管理に特化したクラスについて説明する。ファイルクラス、IOクラス、その他ファイル操作に関連するクラスについて、その概要を解説し、各クラスの構成や使い方を紹介する。	9
I-15-10. Rubyを用いたGUIアプリケーション開発	GUIアプリケーションのインタフェースを構成するウィジェットやイベント処理について説明し、RubyでGUIアプリケーションを開発する際に利用するライブラリであるRuby/TkやRuby-GNOME2の概要と使い方を説明する。	10

I-16. 開発フレームワークに関する知識 I

開発フレームワークとは何か、その基本的な概念、歴史、特徴とメリットを解説する。さらに開発フレームワークを利用して開発を行う際に必要な技術を説明し、実際の開発に役立てることができる知識やノウハウを示す。

習得ポイント	説明	シラバスの対応コマ
I-16-1. 開発フレームワークの歴史、思想と背景	開発フレームワークとは何かを説明し、開発フレームワークの基本理念、目的、発展の歴史、開発フレームワークが登場した背景と思想について解説する。	1
I-16-2. 開発フレームワークの特徴とメリット	開発フレームワークを利用した開発の特徴と、開発フレームワーク利用の効果、メリットについて解説する。また代表的な開発フレームワークを紹介し、それぞれの歴史や背景、特徴、利用上の注意点、開発フレームワークの選択方法、ライセンス形態などを説明する。	1,2
I-16-3. Webアプリケーション構築で利用できるOSSフレームワーク	オープンソースによるWebアプリケーション開発フレームワークを紹介する。Struts、JSF、IBATIS、Hibernate、Seasar 2、Springといった代表的なフレームワークを紹介し、それぞれの開発内容の違いについて解説する。	3
I-16-4. Tomcatの特徴とServlet/JSPによるアプリケーション開発	Java Servlet/JSP コンテナに関する開発フレームワークの代表的なものとして、Tomcatを取り上げ、その位置づけ、特徴とServlet/JSPによるWebアプリケーション開発の具体的な手順を説明する。	4
I-16-5. アプリケーションサーバJBossの機能と特徴	アプリケーションサーバについて、例として代表的なアプリケーションサーバであるJBossを取り上げ、その位置づけ、特徴を解説する。またTomcatとの連携やEJBコンテナ機能、アプリケーションサーバ機能など、JBossの持つ様々な機能を紹介する。	4
I-16-6. ツール実行を自動化するAnt	コンパイルやテスト、バージョン管理などソフトウェア開発に関わるあらゆる処理の実行を自動化するツールAntを説明する。Antの基本処理であるコンパイル、ビルドのXMLによる設定方法を示す。また、Antで自動化できるテスト、ファイル転送、リモートホスト操作、バージョン管理などを紹介する。	5
I-16-7. Javaプロジェクト管理ツールMaven	Javaソフトウェア開発のコンパイル、ビルド、テスト、パッケージングに至る一連の作業の自動化するプロジェクト管理ツールMavenを解説する。スクリプトを記述する必要が少ないことなどAntに比べて優れた点にも触れる。	5
I-16-8. Webアプリケーションのテスト支援ツールCactus	Javaソフトウェアのテスト支援ツールであるJUnitを紹介し、JUnitでテストできないServletやJSPに対応したCactusを説明する。テストコードの記述方法、テストの実行方法を解説する。	5
I-16-9. 要求分析モデル(DOA、OOA、Web MVC)	開発フレームワークによるアプリケーション開発を行う際に実施する要求分析について、DOA(データ中心アプローチ)、OOA(オブジェクト指向分析)、Web MVC(モデル-ビュー-コントローラ)モデルといった各種のモデルの特徴と違いについて解説する。	6
I-16-10. Webアプリケーション開発における実際の作業プロセス	Webアプリケーション開発に関して、プレゼンテーション設計、画面遷移設計、ビジネスロジック設計、データベース設計、コンポーネント間インタフェース設計といった実際の実装作業プロセスを説明する。	6

I-17. 開発ツールに関する知識 I

ソフトウェア開発の基本的な進め方を示し、ソフトウェアを開発する際に活用する一連のツール群について、その機能と使い方を説明する。コンパイラやデバッガといった基本的なツールから、バージョン管理ツール、バグ追跡システムまで、様々なツールの活用法を解説する。

習得ポイント	説明	シラバスの対応コマ
I-17-1. ソフトウェア開発の進め方	ソフトウェア開発に必要なハードウェア、ソフトウェア、ツールの概要を示し、ソフトウェア開発プロセスにおける一連の手順を紹介する。ソフトウェア開発で注意すべきポイントや具体的な開発サイクルなど、実際のソフトウェア開発で活用できるノウハウを紹介する。	1
I-17-2. 各種のツール(コンパイラ、デバッガ、プロファイラ)	ソフトウェア開発環境の全体像を述べ、コンパイラ、デバッガ、プロファイラなど、開発環境を構成する様々な開発ツールの概要、実際の開発における役割りと位置づけ、各ツールの特徴について解説する。	2
I-17-3. 統合開発環境の様々な機能	各種のツールを効果的に組合せ、一連の開発手順をシームレスに提供する統合開発環境について解説する。統合開発環境の歴史や背景、目的、その効果などを紹介し、統合開発環境が持つ特有の機能にも言及する。	2
I-17-4. C言語によるプログラム開発方法	Linuxの開発環境におけるアプリケーションソフトウェア開発の概要として、C言語によるプログラム開発の概要について解説する。Cプログラムの基本を説明し、gccによるコンパイルの手順、生成されるオブジェクトファイルの構造、ライブラリなどについて触れる。	3
I-17-5. バージョン管理ツールの特徴、主なツールと利用方法	変更を重ねて作成するソフトウェア開発の管理に必須のツールであるバージョン管理ツールを紹介する。バージョン管理ツールの機能と特徴、バージョン管理ツールの利用方法を説明し、現在ひろく利用されているOSSのバージョン管理ツール(CVS, Subversion)を紹介する。	4
I-17-6. デバッグの基本(ブレイクポイントとデータのトレース)	デバッガを利用したプログラムのデバッグ作業について説明する。データの変化をトレースしバグの原因を究明する方法の基礎、ブレイクポイントの設定、データの確認方法など、デバッガを利用したデバッグ作業の基本的な手順を紹介する。	5
I-17-7. アプリ、ミドルウェア等における様々なデバッグ手法	プログラムの部分的な実行や条件を変えてのテスト、ウォッチポイントの設定など、デバッガの様々な機能を紹介する。また、アプリケーションやミドルウェアをデバッグする際に活用できる様々なデバッグ手法を説明する。	5
I-17-8. OSの動作に対する追跡手法	OSの動作を追跡する方法について説明する。通常のアプリケーションソフトウェアのデバッグとの違いについて解説し、カーネルデバッガ、デバッグライト、仮想マシン、リモートデバッグなど、OSをデバッグする様々な方法について説明する。	6
I-17-9. バグ追跡システムの目的、機能と利用方法	バグ追跡システムとは何か、その目的、機能、役割りと位置づけを解説する。継続的なデバッグの実現やバグ対応の生産性向上に寄与することを示し、ソフトウェア製品の品質向上に不可欠なツールであることを説明する。	7
I-17-10. OSSバグ追跡システム「Bugzilla」の利用	オープンソースソフトウェアとして提供されているバグ追跡システムの代表的なシステムであるBugzillaについて解説する。Bugzillaの構成、機能を示し、基本的な使い方を説明する。	7

I-18. 統合開発環境に関する知識 I

ソフトウェア開発を効率的に進めるために用意されている統合開発環境について解説する。統合開発環境の歴史、機能、特徴、使い方などを説明する。また OSS の開発でしばしば利用される代表的な統合開発環境のいくつかについて、その特徴、基本的な機能と導入方法を紹介する。

習得ポイント	説明	シラバスの対応コマ
I-18-1. 統合開発環境の概要	統合開発環境の主な機能や統合開発環境を利用するメリット、統合開発環境を利用したソフトウェア開発手順の概要を解説する。	1
I-18-2. さまざまな統合開発環境	OSS開発で用いられる主な統合開発環境の種類と代表例を紹介する。対応言語、対応フレームワーク、開発用途などの特徴と違いを解説する。	2
I-18-3. Eclipseの歴史、開発の背景と特徴	OSS統合開発環境のひとつであるEclipseについて、開発の背景、歴史、支援団体、関連プロジェクト、ライセンスなどについて説明する。さらにそのアーキテクチャや機能、プラグインによる拡張などの特徴を解説する。	3
I-18-4. Eclipseの設定と基本機能	Eclipseを利用するために必要な環境を設定する手順について、パッケージの入手方法から動作環境の整備、プラグインによる拡張方法、関連アプリケーションの設定などについて具体的に説明する。また基本機能について解説する。	6
I-18-5. Eclipseの基本操作	ワークベンチウィンドウ、プロジェクトの作成、コーディング方法、ビルドと実行など、Eclipseの基本的な操作について解説する。またプラグインによる拡張の利用例など、Eclipseの効果的な活用についても触れる。	7
I-18-6. NetBeans IDEの歴史、開発の背景と特徴	OSS統合開発環境のひとつであるNetBeans IDEについて、開発の背景、歴史、コミュニティ、関連プロジェクト、ライセンスなどについて説明する。さらにそのアーキテクチャや機能、ユーザインタフェースなどについて解説する。	4
I-18-7. NetBeans IDEの設定と基本機能、基本操作	NetBeans IDEを利用するために必要な環境を設定する手順について、パッケージの入手方法から動作環境の整備、関連アプリケーションの設定などについて具体的に説明する。また基本機能と基本的な操作について解説する。	8,9
I-18-8. WideStudioの歴史、開発の背景と特徴	OSS統合開発環境のひとつであるWideStudioについて、開発の背景、歴史、コミュニティ、ライセンスなどについて説明する。さらにそのアーキテクチャや機能、MWT (Multi-platform Widget Toolkit)などの特徴を解説する。	5
I-18-9. WideStudioの設定と基本機能	WideStudioを利用するために必要な環境を設定する手順について、パッケージの入手方法から動作環境の整備、作業環境の管理方法などについて具体的に説明する。また基本機能について解説する。	10
I-18-10. WideStudioの基本操作	プロジェクトの作成、アプリケーションウィンドウ、インスタンスの管理、イベントプロシージャ、プログラムのビルドと実行、デバッグ方法など、WideStudioの基本的な操作について解説する。	11

I-19. 暗号化に関する知識 I

OSS アプリケーションのセキュリティを確保するために必須の技術である暗号化について、その基本的な仕組み、手法の種類、各手法の実装方法、インターネット上での応用技術などを解説する。

習得ポイント	説明	シラバスの対応コマ
I-19-1. 暗号化の意義と効果、課題、注意点	OSSにおけるセキュリティの基本概念と全体像を概説する。OSSに求められるリスクとは何か、暗号化でどのような対応が可能か、暗号化処理の種類と利用時の課題、留意点などについて説明する。	1
I-19-2. 共通鍵暗号方式の仕組み	暗号化方式のひとつである「共通鍵暗号方式」の基本概念と仕組み、特徴、利点と欠点、OSSにおける利用状況について説明する。また、米国で標準化されたAES (Advanced Encryption Standard)を紹介する。	2
I-19-3. 公開鍵暗号方式の仕組みと重要性	暗号化方式のひとつである「公開鍵暗号方式」の基本概念と仕組み、特徴、利点と欠点、OSSにおける利用状況について説明する。インターネット利用時の暗号化方式として公開鍵方式がなぜ重要なのかを議論する。	3
I-19-4. ソフトウェア、ハードウェア、通信路における暗号化方法	OSやミドルウェア、アプリケーションで求められる暗号化処理とその実装を紹介する。またハードウェアレベルでの暗号化、ネットワークにおける暗号化にはどのようなものがあるか、その目的と特徴、実装方法を説明する。	4
I-19-5. 電子証明書の仕様、仕組み、役割と必要性	ネットワークにおける各ノードの正当性を証明する電子証明書について、それらの種類、その仕様、仕組み、電子証明書の役割と必要性について述べる。	5
I-19-6. OSSと暗号化、OSSにおける実装事例	様々なOSS活用シーンにおける暗号化の必要性を示し、OSSによる暗号化処理の実装例を、OS、ミドルウェア、アプリケーションのレベルで分類して紹介する。	6
I-19-7. 無線LANに求められる暗号化の仕様、必要性、課題	無線LANにおける暗号化の必要性について述べ、その仕様、特徴、利点と欠点などについて説明する。代表的な暗号化方式であるWEP (Wired Equivalent Privacy)と、さらに強化した暗号化方式のWPA (Wi-fi Protected Access)などを紹介する。	7
I-19-8. 認証の仕組みと目的、実現方法、利点	ネットワーク利用においてユーザや文書の正当性を証明する「認証」の基本的な仕組みと実現方式を解説する。また認証を実現するうえで暗号化をどう利用しているか、その具体的な方法について説明する。	8
I-19-9. Webサーバの暗号化ツール	HTTPによるWebサーバとのやりとりを暗号化する技術の中核をなすプロトコルであるSSL (Secure Socket Layer)について、その概要、仕様、特徴などを解説する。	11
I-19-10. セキュアシェル	リモートホストへのログインや遠隔実行を実現する手段として用意され、インターネットにおける代表的な暗号化通信方法となっているSSH (Secure Shell)について、その概要、仕様、特徴などを解説する。	10

I-20. ネットワークセキュリティに関する知識 I

ネットワークにおけるセキュリティのリスクと、各種リスクに対する対策手法の概要、機能、実装などについて述べる。具体的なセキュリティ要件を説明し、サーバ運用やネットワーク設計におけるセキュリティ実装について実務的な知識を解説する。

習得ポイント	説明	シラバスの対応コマ
I-20-1. ネットワークセキュリティの基本概念、リスクの種類	ネットワークセキュリティの基本概念とセキュリティ確保に必要な機能を解説する。ネットワークセキュリティに関するリスクの種類を説明し、様々なリスクに対するセキュリティ実装技術を紹介する。	1
I-20-2. ネットワークセキュリティに関連する法律とセキュリティポリシー	不正アクセス禁止法、刑法、電子署名法、個人情報保護法など、ネットワークセキュリティに関連する法整備について解説する。また組織におけるセキュリティポリシーの位置づけについて説明し、構成員の情報セキュリティ意識を高める工夫やセキュリティポリシーの定め方について触れる。	1
I-20-3. コンピュータウイルスの種類と特性	コンピュータウイルスについて、その特性、発生する理由、ウイルスの種類、動作原理などを解説する。またどのような経路で感染が拡大するのか、どのような被害が生じる可能性があるのかを説明する。	2
I-20-4. ウィルス対策ソフトウェアの特徴と運用方法	ウィルス対策ソフトウェアの基本的な考え方を紹介する。さらに、クライアント、サーバ、ゲートウェイといったネットワーク上におけるそれぞれのノードで動作するウィルス対策ソフトウェアの特徴と、運用方法、運用上の留意点について述べる。	2
I-20-5. ネットワークに対する攻撃	ネットワークセキュリティに関して、攻撃の種類、攻撃方法の概要を説明する。攻撃手段としては、パスワード推定、設定・プログラムのミスや脆弱性によるセキュリティホールへの攻撃、DoS、ソーシャルエンジニアリングなどについて言及する。	3
I-20-6. 各種サーバへの不正アクセス手法	TCPの仕様に基づいて攻撃する不正アクセスの種類と内容を解説する。具体的には、Telnet、FTP、SSH、SMTP、POP3、IMAPといった各種サーバへの不正アクセスについて述べ、さらにLandやPing of Deathといったセキュリティの弱点をつく攻撃についても触れる。	4
I-20-7. Webシステムへの不正アクセスと対策	Webシステムのセキュリティリスクについて解説し、バッファオーバーフローやDoS攻撃などのWebサービス/Webサーバへの不正アクセスや攻撃の内容と手順、およびそれらに対する対策について説明する。	5
I-20-8. IPプロトコルに対する不正アクセスと対策	IPアドレスの偽造、経路制御の不正、IPソースルーティング、ルーズソースルーティングといったIPプロトコルを悪用した不正アクセスによるセキュリティリスクを紹介し、それぞれの内容と対策について解説する。	6
I-20-9. インターネットセキュリティとネットワークセキュリティの設計と実装方法	TCP/IPネットワークの持つセキュリティリスクと、インターネットで動作するアプリケーションに関するネットワークセキュリティの設計方法、実装方法について解説する。	7
I-20-10. ファイアウォールの仕組みとアクセス制御/フィルタリングの設定方法	ネットワークセキュリティの重要技術であるアクセス制御とフィルタリングについて説明し、その実装であるファイアウォールの機能と設定方法、運用の考え方について解説する。	8

I-21. OS セキュリティに関する知識 I

OS のセキュリティを確保するために必要な機能を、Linux を題材として解説する。ローカルで利用する際のセキュリティ対策を示し、さらにネットワーク経由で利用する際のセキュリティ対策のうち、とくに OS のセキュリティ管理に必要な部分に焦点を当てて説明する。

習得ポイント	説明	シラバスの対応コマ
I-21-1. サーバセキュリティに関する基本概念	現在のOSに求められるセキュリティの基本的な概念と、OSをとりまくリスクの種類とその対策方法など、OSのセキュリティを確保するために必要な事項を解説する。	1
I-21-2. Linuxサーバのローカルセキュリティ対策の基本と設定方法	Linuxサーバの運用時、ネットワーク利用の有無に関わらず必要となるローカルセキュリティ管理について、その基本的な項目の内容と各種の設定方法、セキュリティが保たれているかどうかのチェック方法を説明する。	2
I-21-3. Linuxサーバのネットワークセキュリティ対策の基本と設定方法	Linuxサーバのネットワークセキュリティ管理について、その基本的な項目の内容と各種の設定方法を解説する。またネットワーク上の通信に関してファイバシを保護するための対策についても触れる。	3
I-21-4. ログを用いたセキュリティ管理の基本	Linuxサーバの運用時に得られるログを利用して、サーバセキュリティ管理を行う手法についての基本的な概念と方法を説明する。改ざんが行われたなど、サーバに対する攻撃が行われたことをログを利用してチェックする方法を解説する。	2
I-21-5. Linuxサーバを使用したファイアウォールの設計と導入	ファイアウォールとは何か、ファイアウォールとDMZ (DeMilitarized Zone) の関係、安全なネットワーク構築方法について触れ、さらにLinuxサーバを利用したファイアウォールの構築方法、設定ポリシーの考え方などを説明する。	4
I-21-6. iptablesによるセキュリティ管理方法	Linuxサーバによるファイアウォールを実現するiptablesの目的や役割について概説する。またiptablesの具体的な設定方法を紹介し、その運用手順について解説する。	4
I-21-7. セキュリティを考慮したLinuxサーバの適切な設定	サーバソフトウェアのインストール方針やサービス提供方針など、Linuxサーバにおけるセキュリティ管理の基本的な方針について説明し、さらにサーバ上で十分なセキュリティを確保するために必要な作業と留意点を示す。	5
I-21-8. セキュアシェルの利用、その基本と応用	リモートからの安全なアクセスを実現するSSH (Secure SHell)について解説する。SSHの基本的な動作とOpenSSHの導入、SSHに関する各種の設定などを説明し、さらにSSHの応用としてリモートからのコマンド実行やTCPポートフォワーディングにも言及する。	6
I-21-9. SSLによる仮想専用ネットワーク(VPN)の設定	Webの暗号化通信技術SSLを利用して仮想専用ネットワーク(VPN)を実現するSSL-VPNの仕組みについて解説する。利用できるアプリケーションが限定されるが、導入が容易であることに言及する。	7
I-21-10. DNSに関するセキュリティ対策	DNS (Domain Name System)の基本的な動作とDNSに求められるセキュリティ要件を解説する。標準的なDNSサーバの実装であるBINDを紹介し、BINDにおけるセキュリティ対策について述べる。	8

I-22. RDB に関する知識 I

現在におけるデータベースの主流である関係データベース(RDB)に関して、その基本的な構成、機能と役割、特徴を解説する。また実際の開発や運用に必要な設計手法や正規化手法、分析手法について説明を加える。

習得ポイント	説明	シラバスの対応コマ
I-22-1. データベースの意義、位置付けと利用方法	データベースを利用することのメリット、データベース管理システム(DBMS)の位置づけとOSSの実装、データベースの構造など、データベースを利用するための基本的な概念について説明する。	1,2
I-22-2. RDBMSの特徴、関係モデルとSQL	データ構造モデルとしての関係モデルを説明したうえで、関係モデルに基づく関係データベース管理システム(RDBMS)を説明し、データベース言語SQLの概要を紹介する。	1,2
I-22-3. 集合演算の概念	関係データベースの基礎となる関係モデルについて、その演算に関する概念を説明する。和、積、差、直積といった通常の集合演算と、射影、結合、選択、商といった関係代数独自の演算について解説を行う。	4
I-22-4. トランザクションの概念、排他制御とロック	データベースの特性とデータベース設計に不可欠なトランザクションの概念と特徴について解説し、確実なトランザクションを実現するための排他制御とロックの概念およびその実装方法を説明する。	3
I-22-5. 関係データベースの構成要素	関係データベースの構成要素である表、行、カラムなどの意味について説明する。またレコードを一意に特定するための主キー、複数属性からなる主キー、外部キーなど、リレーションスキーマに関するいくつかのトピックを紹介する。	4
I-22-6. 整合性を保つ方法と各種の演算	関係データベースを構成する各要素の間で整合性を保持する方法や、整合性が保たれるための条件を説明する。またリレーショナルデータモデルに対して行われる各種の演算を、演算結果の例を示しながら分かりやすく説明する。	4
I-22-7. DOA(データ中心分析)の意義、目的、内容	従来のプロセス中心設計に代わるデータベース設計手法として着目されているデータ中心分析DOA(Data Oriented Approach)の概要と、その意義、目的、具体的な手順について概説する。また関連する話題としてデータモデルおよび3層スキーマについても触れる。	5
I-22-8. データベースの設計手順	データベース設計方法と設計手順を解説する。業務分析、概念設計、論理設計、物理設計とデータベース設計の手順を進める中で、正規化が重要な役割を担うことを示す。また分析の手順としてトップダウンアプローチおよびボトムアップアプローチがあることを説明する。	6
I-22-9. ERモデルの具体的な考え方と作成手順、記述法	関係データベース設計の重要な表現法であるERモデル(Entity-Relational Model)について、基本的な考え方、データの表現方法や構成要素、具体的な作成手順を解説する。またERD(Entity-Relational Diagram)を用いたデータ設計手順について説明する。	7
I-22-10. データベース正規化の具体的な考え方と手順	関係データベース設計の必須技術である「正規化」について、具体的な考え方と正規形の種類、関数従属といった基本的な概念を説明する。また正規化の手順を示し、各手順におけるポイントや正規化を崩すケースなどについて述べる。	8

I-23. RDB システム管理に関する知識 I

関係データベースの運用管理機能とその管理方法について、具体的な内容と管理のための知識を解説する。バックアップとリカバリ、セキュリティに対する配慮、データベース運用時の動作監視といった基本的な項目について説明する。

習得ポイント	説明	シラバスの対応コマ
I-23-1. データベース運用管理の目的、方針、主要な問題点と対策	データベースの運用管理が必要になる状況を示し、運用管理の必要性と目的、運用の方針、データベース運用時に遭遇する主要な問題点やリスクとその対策について説明する。	1
I-23-2. データベース運用管理の基本的な項目と作業内容	データベース運用管理の基本的な項目と作業内容を解説する。定期的に行わなければならない作業、継続した利用の際に発生する可能性がある問題に対する作業など、データベース運用に不可欠な作業項目を順序だてて説明する。	1
I-23-3. 運用設計の概要	データベース運用設計作業の内容について述べる。最適なデータベース運用設計のアプローチ方法、アプリケーション設計への反映方法やデータベース運用計画への反映方法、監視計画の策定方法について説明する。またデータベース運用設計作業で検討すべき項目について、その作業内容と手順を説明する。	3
I-23-4. 障害の種類と障害からの復旧方法	データベース運用時に発生する障害の種類とその内容を示す。これらの障害に対して事前にとっておくべき対策と、障害が発生した際に復旧作業としてやらねばならないことの概要を説明する。	2
I-23-5. データベースのバックアップ	一般的なデータベース運用作業における基本項目のひとつとして、障害回復に備えるためのバックアップ作業について説明する。バックアップ作業の必要性、バックアップの種類、バックアップ作業手順などについて述べる。	2
I-23-6. データベースのリカバリ	一般的なデータベース運用作業における基本項目のひとつとして、障害発生時のリカバリ方法について述べる。データベースのリカバリ作業の必要性、具体的な手順を示し、リカバリ作業時の留意点などについて説明する。	5
I-23-7. ログの取得と復旧	データベース運用管理作業において復旧時に取るべき対策のひとつとして、ログの取得について説明する。取得したログの見方と障害の原因追及方法、ログを利用したリカバリ方法などについて解説する。	2,4
I-23-8. データベースのユーザ管理とアクセス制御	データベースにアクセスできるユーザを管理する方法と、アクセス制御の概念について、ユーザ管理およびアクセス制御を実現するための、基本的な作業手順を示す。またこれらに関する作業上の留意点を説明する。	4
I-23-9. データベースセキュリティの内容と留意点	データベース運用時に配慮すべきセキュリティの概要を紹介し、実際の運用作業において実施しなければならないこと、注意すべきポイントについて解説する。また実際にセキュリティ侵害が発生した際の対処方法について説明する。	4
I-23-10. データベース動作環境の管理計画と管理作業	データベースを正常に動作させるために実施しなければならない作業について、バックアッププランの策定、リカバリ方法の決定、データベース動作環境の監視といった一連の作業手順についてまとめる。	5

I-24. 組み込みシステムに関する知識 I

組み込みシステムの基本的な構造と活用方法、組み込みシステムを構成するハードウェア、ソフトウェア、OS など様々な要素の役割や特徴について解説する。

習得ポイント	説明	シラバスの対応コマ
I-24-1. 組み込みコンピュータシステム	組み込みコンピュータシステムとは何か、基本的な概念を説明する。組み込みコンピュータシステムにおけるハードウェア、ソフトウェア、ネットワークの全体的なアーキテクチャを解説する。	1
I-24-2. 組み込みシステムとOSSの関係、活用事例	組み込みシステムにおける数多くのOSS活用事例を紹介し、組み込みシステムの構築にOSSの活用が有効であることを説明する。また組み込みシステムにおけるOSS活用時の留意点についても解説する。	1
I-24-3. 組み込みシステムの制限と設計方法論	組み込みシステムにおいてはハードウェア、ソフトウェア、ネットワークのリソースが限られていることを示し、組み込みシステムにおける各種の制限について説明する。さらに、その条件下でうまくシステムを構築するための設計方法を解説する。	2
I-24-4. 組み込みコンピュータの要件	組み込みコンピュータの基本構成と特徴について述べ、組み込みコンピュータに求められる要件として、接続性、信頼性、効率性、移植性などがあることを示し、それらの内容について説明する。	3
I-24-5. 組み込みシステムのCPU、OS、ネットワーク機能	組み込みシステムに利用される代表的なCPU、OS、ネットワーク形態を紹介する。またそれらの特徴を解説し、組み込みシステム特有の課題や対応方法(低消費電力など)についても述べる。	3
I-24-6. 組み込みシステムにおけるハードウェアの基本構成	組み込みコンピュータハードウェアの基本構成と役割、特徴を説明する。中心となるCPUのアーキテクチャ、システムバスのアーキテクチャ、バスを介したデータのやりとりとCPUにおける演算処理の実際等について概説する。	4
I-24-7. 組み込み向けCPUのアーキテクチャ	組み込みコンピュータの中核となるCPU内部のアーキテクチャについて、詳細な説明を行う。また命令セットやメモリとの接続、メモリ管理方式等についても解説を加え、さらに、代表的な組み込み向けプロセッサのいくつかについて、そのアーキテクチャを紹介する。	5
I-24-8. 組み込み向けソフトウェアの基本	組み込み向けソフトウェアの役割と特徴を説明する。組み込み向けソフトウェア処理の基本として、タスクとスケジューリングを説明し、並行処理のアーキテクチャに関する基本的な方式とタスク管理方法などに関する話題にも触れる。	6
I-24-9. 組み込みOSカーネルの機能	組み込み向けOSのカーネルに関して、その機能と役割、特徴を解説する。組み込みOSカーネルに求められる機能や制御対象、タスク制御に関わる組み込み特有の問題、排他制御の必要性とその具体的な実装方法などを説明する。	7
I-24-10. リアルタイムシステムの処理方法と設計	組み込みシステムの特徴のひとつであるリアルタイム性について説明する。リアルタイムシステムが備える機能と、リアルタイム処理の特徴について説明し、さらに、アプリケーションのリアルタイム設計方法、時間制御と優先度といったトピックを紹介する。	8

I-25. 組み込み開発環境に関する知識 I

組み込みシステムの開発手法と組み込みシステムを開発するための開発環境について、その基本構成と特徴を解説する。さらにデバッガやエミュレータを利用した組み込みシステム開発の実際を、ツールの紹介を交えて具体的に示す。

習得ポイント	説明	シラバスの対応コマ
I-25-1. 組み込みシステムの開発の概要	組み込みシステムを開発するための開発環境についてその基本構成と特徴を概説し、必要なハードウェア、ソフトウェア、ツール群を紹介する。	1
I-25-2. 組み込みLinuxによる実装とその開発	組み込みLinuxを用いた組み込みシステムの実装について説明する。ブートローダからLinux実装までの基本構成や、実際のアプリケーション開発方法、プログラミングからROM化までの手順について、組み込みLinuxシステムを題材に解説する。	1
I-25-3. 組み込みシステムの開発環境	組み込みシステムの開発環境について、全体像とその構成、特徴や開発手法などを説明する。ターゲットOS、言語、ミドルウェア、プログラミング方法、エミュレーション、デバッグといった話題について触れ、統合開発環境が持つ管理機能についても言及する。	2
I-25-4. 組み込みシステムの開発技術	組み込みシステム開発の全体像について、システム全体の設計段階、ハードウェア設計段階、ソフトウェア設計段階からそれぞれの開発を経てシステム提供に至るまでの手順を説明する。またそれぞれの手順で利用される開発環境やツールを紹介し、それらに求められる機能についても解説する。	3
I-25-5. 組み込みソフトウェアのデバッグの基本手順	デバッグの基本であるプログラムのトレースにおいて、組み込みシステムならではの特徴について説明し、トレースの基本手順を解説する。またリアルタイム処理に係るトレースの留意点について説明する。	4
I-25-6. デバッガを利用したプログラムのデバッグ	デバッガを利用したデバッグ環境の全体像と構成、特徴を説明する。デバッグ環境構築上の留意点や組み込みシステムを対象としたデバッグ特有の特徴や留意点などについて説明する。	5
I-25-7. エミュレータの利用方法	エミュレータとは何か、その基本構成と特徴、役割を説明する。またエミュレータを用いたデバッグやプログラム実行の検証、解析方法について解説を加える。	6
I-25-8. ICEを利用したデバッグ環境	組み込みシステムに特有のハードウェアエミュレーションについて説明し、ICE (In Circuit Emulator)の必要性和目的、使い方、ICEを利用したハードウェア関連のデバッグ方法を解説する。また組み込みLinuxに対応した代表的なICEを紹介する。	6
I-25-9. ツールチェーンを利用したデバッグ	ツールチェーンとは何か、その機能と特徴、利点と欠点などについて説明する。またエンディアンの問題やクロスコンパイル環境といった組み込みシステム開発特有の話題について触れ、ツールチェーンを利用したプログラム開発とデバッグの手順を紹介する。	7
I-25-10. 実機レベルのデバッグとトラブルシューティング	組み込みシステムを対象としたデバッグの実際に関して、デバッガや開発環境の制限事項で留意すべき点や、チェックすべき対象、様々な状況下でのトラブルシューティング方法、トラブル解決に利用できるツールなど、様々な話題を紹介する。	8

I-26. 組み込みアプリケーション開発に関する知識 I

組み込みシステムで動作するアプリケーション開発について、アプリケーションの基本的な設計と特徴を解説し、さらに資源管理や資源の共有、割り込みに関する処理などについての具体的な実装について説明する。

習得ポイント	説明	シラバスの対応コマ
1-26-1. コンテキストスイッチの仕組み	複数の処理(タスク)を実行するコンテキストスイッチの仕組みを解説する。OSが処理の切り替えを管理するプリエンブション方式の実装方法を説明し、割り込みによるコンテキストスイッチにも触れる。	1
1-26-2. 非同期処理と同期処理の実装パターンと特徴	非同期/同期の概念について触れ、その実装パターンと設計について内容と特徴を説明する。また実行モードの種類や割り込みの優先順位といった具体的な事項について説明する。	2
1-26-3. タスク優先順位制御とカーネルによる時間管理方法	タスクの優先度とその制御仕様について、実装パターンと設計内容、特徴、手順を説明する。実際のタスク制御を行うカーネルが時間制御方法としてどのような管理手法を用いているかについても触れる。	3
1-26-4. システムリソースの配分の仕組み(セマフォ、キュー、等)	組み込みシステムにおいては、システムリソースが限定されているため様々な工夫が必要になる。各アプリケーションが限られたシステム資源を効率的に共有する方法について、セマフォ、キュー、といった具体的な実装方法を踏まえて、その内容と特徴を説明する。	4
1-26-5. システムリソースの共有の仕組み(共有ファイル、デッドロック回避、等)	複数の組み込みアプリケーション同士でシステムリソースを共有する方法を、共有エリア・共有ファイル、カーネルによる各種のサービス、デッドロックの回避、割り込みディスパッチなどの具体的な実装を挙げて説明する。	5
1-26-6. CPUによるメモリ資源の管理方法	システムリソースのうち、とくにCPUが介在するメモリ空間の管理方法に焦点を当て、ベースアドレス方式、セグメント方式、ページング方式、バンクメモリ方式等、その具体的な実装方法を説明する。	6
1-26-7. メモリの共有制御とプログラムパーティション	複数のアプリケーションがメモリを共有するための制御方式として、固定長方式や可変長方式があることを示し、それぞれの特徴、利点と欠点について説明する。さらにプログラムパーティションの管理、ガベージコレクションといった話題にも言及する。	6
1-26-8. プログラム資源の管理、複数タスクによる同時利用	プログラム資源の管理、有効活用の方式、実装方法について解説する。また複数タスクの同時利用方法として、処理待ちキュー、タスクの優先順位、入出力装置による待ち時間の勘案といったタスク管理も説明する。	7
1-26-9. 入出力資源の管理方法、同時利用方法	実デバイスを操作することが多い組み込みシステムにおいては、とくに入出力装置に対する待ち時間を有効に利用することが求められる。ここでは、入出力待ち時間を有効活用するための方法と、各種の制約、特徴、実装上の留意点などについて述べる。	8
1-26-10. リソースに対する優先順位の設定、割り込みの優先順位	組み込みアプリケーションの非同期性について解説し、リソースの排他的利用、リソース利用に対する優先順位設定、割り込みの優先順位、デバイスドライバ処理の優先順位など、各種の順位決定処理を説明する。	8

I-27. 組み込みシステム最適化に関する知識 I

システムの性能を最適化するための方法としてのマルチプロセッサ導入や、リアルタイムシステムの特性を活かした最適化、ソフトウェア最適化や最適化のためのシステム評価方法など、組み込みシステムの最適化に関する様々な技術について解説する。

習得ポイント	説明	シラバスの対応コマ
I-27-1. マルチプロセッサシステムの種類	マルチプロセッサシステムについて、その基本概念と特徴、利点などを示し、従来からの分類方法であるSISD/SIMD/MISD/MIMDといった種別や、新しい分類方法であるSMP (Symmetric Multi Processor)、クラスタ構成、MPP (Massively Parallel Processor)などを紹介する。	1
I-27-2. マルチプロセッサシステム活用上の留意点、構成例	タスクのスケジューリング、プロセッサ間の同期と通信、ハードウェアの制限など、マルチプロセッサシステムを活用する際に留意すべき事項を説明する。またマルチプロセッサシステムの具体的な構成例を紹介し、その特徴について説明する。	1
I-27-3. マルチプロセッサによるシステムの最適化	性能の最適化方法、CPU高速化、入出力方法に関する工夫、低電力処理など、マルチプロセッサを使用したシステムを最適化する手順や配慮すべき項目について解説する。	2
I-27-4. リアルタイムシステムの設計と留意点	リアルタイムシステムの設計に必要な要素を解説する。具体的には、モジュール分割、処理分割による最適化、割り込み対応、処理時間・応答時間の見積り、処理待ちキュー設計、デバイスに対する高速応答処理の実装などについて触れる。	3
I-27-5. タスク分割による性能向上	リアルタイムソフトウェアを利用したシステムの性能向上、最適化のためのソフトウェア作成方法を解説する。特に適切なタスク分割によるタスクの同期、並行動作の実現など、関連する話題を説明する。	4
I-27-6. タスク関連設計によるリアルタイムシステムの最適化	タスク関連設計を中心とした最適化のアプローチについて解説する。タスク関連設計の基本的な考え方と特徴について説明し、さらに、タスク関連図の作成方法、タスク優先順位の決定、起動タイミング、タスク間の同期、排他制御といった項目について説明する。	4
I-27-7. リアルタイム処理に対する評価項目と留意点	リアルタイムシステムの性能評価について、全体のパフォーマンス、処理のスループット、全体の優先順位、デバイスドライバの処理方法などの観点からみた評価項目と評価時の留意点を説明する。またタスク分割の設計評価についても解説する。	5
I-27-8. CPUの性能指標と評価方法	CPI (Clock Cycles Per Instruction)、MIPS (Million Instructions Per Second)など、CPUの性能を評価する指標を紹介する。ただし様々なシステムの利用状況においては、これらの指標だけでは不十分であり個別の最適化手法を適用しなければならないことについても言及する。	6
I-27-9. ソースコード最適化によるシステム性能向上	ソフトウェア最適化によるシステムの性能向上について、基本的な概念と設計方針を解説する。ソフトウェア最適化の基本手段である冗長なコードの排除やループの展開、インライン化、変数のレジスタ割り当てなど個別の手法を紹介し、またソースコードレベルの最適化の手順についても説明する。	6
I-27-10. プログラムモジュール配置による最適化	プログラムモジュールの配置方法に係る最適化手法を解説する。プログラムモジュールのメモリへのローディング方法、リンク方式、常駐化など具体的な方法について説明する。	6

7. OSS 基本知識の学習ガイダンス

OSS 基本知識 26 科目の各々について、学習ガイダンス資料を策定した（別冊（26 分冊）参照）。各科目の学習ガイダンス資料は、以下の構成となっている。

- 表紙（2 ページ）
 - 科目の概要
 - 習得ポイント（本報告書 6 章で示したもの）
 - IT 知識体系との対応関係（本報告書 5.1 節で示したもの）
 - OSS モデルカリキュラム固有の知識（本報告書 5.1 節で示したもの）

- 習得ポイント別の解説資料（10 ポイント×2 ページ）
 - 学習の要点（習得ポイントを具体化した学習内容の要点）
 - 解説図（学習の要点を図表化したもの）
 - 解説（学習の要点に関する具体的な解説）

本学習ガイダンス資料は、大学・高専・専修学校、研修機関、企業等における OSS 教育現場において以下の用途で用いることを想定している。

- 「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。

OSS 基本知識の学習ガイダンスに関する調査
OSS 基本知識の学習オリエンテーション
調査報告書【概要版】

独立行政法人 情報処理推進機構

Copyright(c) Information-technology Promotion Agency, Japan. All rights reserved 2008