

調査 5 モデルカリキュラムの提言 コースウェア

26. 組み込みアプリケーション開発に関するスキル

I. 概要	組み込みアプリケーションの制御パターンやアーキテクチャを、実際の実装事例の研究を交えながら、留意点や特徴を学ぶ。
II. 対象専門分野	職種共通
III. 受講対象者、 受講前提	本カリキュラムの基本となる組み込みコンピュータ科学基礎、ソフトウェア、ハードウェア基礎(ITSS レベル1程度)を習得・経験しているレベルの知識を有すること。
IV. 学習目標	<ul style="list-style-type: none"> ● 組み込みアプリケーションはどのような制御から成り立っているのか、そのパターンと構成、制御のポイントを理解する。 ● 組み込みアプリケーションの構成、アーキテクチャはどのようなものがあるかを理解する。 ● 組み込みアプリケーションの実際の設計内容と留意点を理解する。
V. 使用教科書、 教材等	<p>『リアルタイム組み込み OS 基礎講座』 Carolone Yao/Qing Li 著、宇野みれ/有限会社ハラパン・メディアテック訳、宇野俊夫/有限会社ハラパン・メディアテック監修、翔泳社刊</p> <p>『Java モバイルアプリケーション J2ME で実現するユビキタスコンピューティング』立川敬行著、ソフトリサーチセンター刊</p> <p>このほか、オリジナル教材を作成するものとする。</p>
VI. 習得スキルの 評価方法	講義終了後の受講レポート、定量アンケート、知識確認ミニテスト、演習問題の取り組み状況を総合的に判断して評価を行なう。
VII. カリキュラム の構成	<p>レベル 1 第 1 回～第 8 回</p> <p>レベル 2 第 9 回～第 15 回</p>

講座内容

第1回 タスクとコンテキスト(講義 90分)

オープンソースアプリケーション開発のためのタスク処理とコンテキストについて、開発の理由と特徴、設計内容と手順を学ぶ。

(1) コンピュータアーキテクチャとは

1. コンテキストスイッチング
 - ・ タスクスイッチング処理
 - ・ 資源によるスイッチング
 2. スwitchingの条件と実装仕様
 - ・ プリエンプション方式の効率よい実行
 - ・ カーネル処理／割り込み処理による切替
 - ・ カーネルが制御対象とする処理の監視と切替
- 実行中
実行可能
待機など

(2) カーネルコンテキスト

1. 割り込みコンテキスト
2. アプリケーションのコンテキスト分解
3. コンピュータアーキテクチャの意味と発展の歴史、必要性
4. ハードウェアのアーキテクチャの特徴と役割を理解

第2回 非同期と同期の設計仕様(講義 90分)

組み込みアプリケーション仕様として必須の非同期と同期の設計仕様について、実装仕様のパターンと設計内容、特徴、設計手順を学ぶ。

(1) 非同期と同期の設計

1. 非同期と同期のタスク制約
2. システム全体のコンテキストの設計
3. コンテキストインタフェースの実行設計

(2) タスクの構成資源

1. プログラムとタスクの組み合わせ方
2. タスクマッピング
3. 実行モードと起動方法
4. カーネルモード
5. ユーザモード
6. モードと起動

(3) 実行環境

1. コンテキストの実行環境
2. 処理の優先順位
 - ・ 割り込みの優先順位
 - ・ 外部割り込み
 - ・ NMI

第3回 タスクの優先度とその制御仕様(講義 90分)

組み込みアプリケーション仕様として必須のタスクの優先度とその制御仕様について、実装仕様のパターンと設計内容、特徴、設計手順を学ぶ。

(1)タスクの優先順位

1. 追い越し
2. プログラムのリエントラント性と優先度
3. 同期／排他制御と優先度

(2)資源の管理

1. 処理の階層化による資源配分
 - ・ 処理の階層化とカーネル
 - ・ 暗黙的資源配分と明示的資源配分
2. タスクの資源配分
 - ・ 資源配分技術の応用
 - ・ 制御機能の独立
 - ・ すべてをメインループ処理と割り込み処理で構成する場合

(3)カーネルによる時間管理

1. タイムスライス方式
2. イベントドリブン方式
3. フィードバック待ち行列

第4回 組み込みアプリケーション間の資源配分技術(講義 90分)

組み込みアプリケーション同士で限定されたシステムリソースをいかに共有するかについて、実装仕様のパターンと設計内容、特徴、設計手順を学ぶ。

(1) 資源配分技術

1. 配分すべき資源の例
 - ・ MPU リソース
 - ・ メモリリソース
 - ・ ネットワークリソース
 - ・ 制御時間
2. クリティカルセクション
3. 同期制御と排他制御
4. ハードウェアによる方法
5. マルチ MPU の排他制御
6. ソフトウェアによる方法
 - ・ デッカーのアルゴリズム
 - ・ ランポートのアルゴリズム

(2) カーネルの提供する手段を使う方法

1. セマフォ
 - ・ セマフォの特徴
 - ・ バイナリセマフォとジェネラルセマフォ
2. キューイング技法
 - ・ リンク型キューの例
 - ・ タスク間の通信

第5回 組み込みアプリケーション間のリソースの共有技術(講義 90分)

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具体的な実現方法について学ぶ。

(1) 共有エリア・共有ファイル

1. 共有エリア
2. 共有ファイル

(2) カーネルによるサービス

1. メールボックス
2. パイプ
3. ランデブ

(3) デッドロック

1. リソース管理
2. MPU 資源の管理
3. スケジューリング

(4) 割り込みスケジューリング(ディスパッチング)

1. タスクステータス
2. タスク優先順位

第6回 リソース有効活用のアーキテクチャ(講義 90分)

リソース有効活用の方式、実装方法を理解する。具体的な実装内容を理解する。

(1)MPU

1. タイムスライス
2. ラウンドロビン方式
3. イベントドリブン

(2)メモリ資源の管理

1. メモリ空間の管理
 - ・ 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
2. バンクメモリ
3. メモリプールの管理

(3)メモリ共有制御

1. 可変長と固定長
 - ・ 固定長ブロック管理方式
 - ・ 固定長ブロック管理
 - ・ 可変長エリア管理方式
 - ・ 可変長ブロック管理
 - ・ 可変長メモリプール
 - ・ フラグメンテーションの問題
2. メモリプールの空きエリア管理

(4)プログラムパーティション

1. 管理方法
2. ガーベジコレクション
3. プログラムパーティションとタスク優先順位

第7回 プログラム資源の有効活用技術(講義 90分)

プログラムリソース有効活用の方式、実装方法を理解する。具体的な実装内容を理解する。

(1)プログラム資源の管理

1. リューザブルとリエントラント
 - ・ 必要性
 - ・ セグメント方式
 - ・ オーバレイ方式
 - ・ オーバレイセグメント
2. ロールイン/ロールアウト
3. 動的メモリ管理機能
4. ワーキングセット

(2)複数タスクの同時使用

1. 処理待ちキューの作り方
 - ・ 処理依頼の到着順
 - ・ タスクの優先順位順
 - ・ 入出力装置/他のマイコンとの関係の効率順
2. 処理待ちキューの要件

第 8 回 入出力待ちリソースの有効利用(講義 90 分)

入出力待ち時間の有効利用方法と制約、特徴を理解する。

(1) 入出力待ち時間の有効利用

1. 入出力制御
 - ・ 入出力資源の特徴と効果的な使用方法
 - ・ 入出力資源の排他性
 - ・ ハードディスクに対する入出力要求のキュー
2. 割り込みの利用
3. 入出力資源の同時使用方法
 - ・ 制御方法
 - ・ リエントラント性
 - ・ 同時並行処理できるデバイス単位にリエントラント性を保証する

(2) 非同期性

1. 処理待ちキューと同期方法
2. 入出力資源の逐次使用(排他的使用)
3. IOQ の処理
4. IOQ と同期

(3) 処理の優先順位

1. IOQ の優先順位
2. デバイスドライバの処理の優先順位
3. 割り込みの優先順位

第9回 入出力資源管理(講義 90分)

組み込みアプリケーション設計における最適な入出力資源管理プログラムの構造と設計について、その内容と設計の方法を理解する。

(1) 入出力設計の重要項目

1. 入出力を実装する場所
 - ・ OS
 - ・ HW
 - ・ 個別デバイスインタフェース
2. デバイスドライバの機能と設計
 - ・ OS機能とするもの
 - ・ タスク処理で実現するもの
 - ・ 形態別の設計内容と留意点

(2) デバイスドライバの基本機能

1. デバイスドライバの機能分割
 - ・ リクエスト部
(キック部、ストラテジ部)
 - ・ 割り込み処理部
 - ・ 異常監視部
 - ・ デバイスドライバの動作モデル
2. 利用プログラムインタフェース
 - ・ リクエスト部
 - ・ 割り込み部
 - ・ デバイスドライバの割り込み部の分割
 - ・ 異常監視

(3) ファイル入出力の構成

1. ファイルシステムの機能概要
 - ・ ファイルシステムのリエントラント性
 - ・ ファイルシステムの提供方法
2. ファイルシステムタスク
 - ・ デバイス独立性とアクセス単位
 - ・ 領域の割り当て
 - ・ 連続したエリアへのファイル割り当て
 - ・ ブロック単位のファイル割り当て
 - ・ 読み込み／書き出し単位
 - ・ デブロッキング

第 10 回 J2ME の仕様(講義 90 分)

組み込み開発のオープンソース開発技術である J2ME(Java 2 Micro Edition)の仕組みと機能、動作仕様などについて理解する。

(1)J2ME の概要

1. J2ME の歴史
2. J2ME の誕生と普及
3. J2ME の構成

(2)J2ME の基本仕様

1. コンフィグレーションとプロファイル
 - ・ コンフィグレーション
 - ・ プロファイル
2. CLDC(Connected Limited Device Configuration)
 - ・ 概要
 - ・ 環境条件
 - ・ Java アプリの動的配信
 - ・ API(クラスライブラリ)
3. CDC(Connected Device Configuration)
 - ・ 概要
 - ・ 機能条件
 - ・ アプリケーションモデル
 - ・ サポートするプロトコル
 - ・ サポートするロケールエンコード
 - ・ セキュリティ
 - ・ API(クラスライブラリ)
4. MIDP(Mobile Information Device Profile)
 - ・ 概要
 - ・ 環境条件
 - ・ アーキテクチャ
 - ・ ネットワーク
 - ・ 永続データ領域
 - ・ アプリケーション実行管理
 - ・ ユーザインタフェース
 - ・ API(クラスライブラリ)
 - ・ バージョン 2.0
5. FP(Foundation Profile)

- ・ 概要
 - ・ サポートされるプロトコル
 - ・ API(クラスライブラリ)
6. MIDlet
- ・ 概要
 - ・ 実行環境
 - ・ ライフサイクル
 - ・ MIDlet スイートの配信
7. Xlet
- ・ 概要
 - ・ ライフサイクル
 - ・ XletContext
 - ・ Xlet 間通信 (IXC)

第 11 回 VM の概要のその活用(講義 90 分)

組み込みアプリケーション設計開発における VM の特徴、活用方法を理解する。

(1)組み込み Java の実行環境

1. 組み込み Java VM
2. 組み込み Java VM の構成と動作仕様

(2)KVM

1. 開発の目的
2. JVM との違い
3. 仕様
4. 適用製品と機能

(3)オープンソース Waba

1. Waba VM
2. Waba ランタイム
- 3.ブリッジクラス
4. Waba 基本クラス
5. アプリケーションモデル
6. ユーザインタフェース
7. 入出力

(4)セキュア VM

1. 機能
2. 実装仕様

第 12 回 高信頼性の実装(講義 90 分)

組み込みアプリケーションにおける高信頼性の実装方法、活用事例を中心に紹介し、その実際を理解する。

(1) 高信頼性の実装

1. 汎用性と高信頼性
2. 高信頼性
 - ・ フォールトアボイダンス (Fault Avoidance)
 - ・ フォールトトレランス (Fault Tolerance)
 - ・ Nバージョンプログラム
 - ・ リカバリブロックスキーム
3. 高信頼化のための方式
 - ・ 誤り (error)
 - ・ 誤り検出
 - ・ パリティチェックによる誤り検出の効果
4. 誤り検出のための符号化
 - ・ 組織符号
 - ・ バーガー符号
 - ・ M-out-of-N 符号
5. ウォッチドッグタイマ
6. オーディタ

(2) 誤り回復

1. バックワードエラーリカバリ方式
2. チェックポイント技法
3. ジャーナル技法
4. リカバリキャッシュ機構
5. リカバリ領域

第 13 回 組み込みアプリケーションの実装事例研究(ワークショップ 90 分)

多様な組み込みアプリケーションの実装仕様の事例研究を行い、設計方針や機能分割、実装方法を学ぶ。

- (1) 自動車ボデー制御
 - 1. 複数マイコンとのネットワーク連携、センサー制御
- (2) 携帯電話アプリケーション分割
 - 1. 限定された動作環境における機能配分
- (3) 産業ロボット動作制御
 - 1. 精密動作制御
- (4) カーナビゲーションシステム制御
 - 1. マルチメディア制御

第 14 回 クリティカルなマイコンアプリケーションを活用した設計ケースワーク(ワークショップ 90 分)

マイコンアプリケーションの設計ワークショップを行う。ロボットのアプリケーションを設計する。ハードウェア、動作プラットフォーム前提で制約を理解し、組み合わせてアプリケーションを設計するケースワークを実施する。ハードウェア設計をもとに設計を行う。

- (1) 仕様の提示
- (2) 前提条件の把握(組み込み Linux、(組み込み)Java での実装)
- (3) 上記の特性を活かしたソフトウェアの配置と役割分担
- (4) ハードウェア構成とその機能の分析
- (5) 機能設計
 - 1. クラス図作成
 - 2. 状態遷移図作成
- (6) 制御設計、タイミングチャート作成

***実際の設計事例をもとに設計を行い、それぞれ比較し、技術の推移や OSS そのものの普及度合いも評価する。**

第 15 回 組み込みアプリケーションソフトウェアの実装(ワークショップ 90 分)

第 14 回で設計したシステム基盤を構築する。J2ME アーキテクチャを用いて、第 14 回の設計の中の制御仕様の一部の設計を行う。第 13 回の設計内容の一部モジュールを実装する。

- (1)プログラミング
- (2)検証
- (3)仕様・性能評価
 - 1. 受講生間でプログラミングのレビューと評価を行う。

以上