

## 調査 5 モデルカリキュラムの提言 コースウェア

### 25. 組み込み開発環境に関するスキル

I. 概要	GUIプログラミング(GTK+, QT, Qtopia)などのソフトウェアが動作する基盤となるコンピュータシステム、ハードウェア、アーキテクチャについて、その種類と特徴、動向を理解し、OSS を用いて開発するためのハードウェアや基盤に必要な一連の基礎知識と活用技術の概要について学ぶ。
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"><li>● 組み込み開発環境を構成するソフトウェア、ハードウェア、ツールの概要と開発やデバッグの進め方を学ぶ。</li><li>● 組み込み開発環境を構成するツールの概要と開発やデバッグの進め方を学ぶ。</li><li>● 組み込み開発環境を使用する開発やデバッグの進め方を学ぶ。</li></ul>
V. 使用教科書、 教材等	『わかりやすい組み込みシステム構築技法ーソフトウェア編ー』 永井正武監修、澤田勉著、共立出版刊 上記のほかにオリジナル教材を作成するものとする。
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う。
VII. カリキュラム の構成	レベル 1 第 1 回～第 8 回 レベル 2 第 9 回～第 15 回

## 講座内容

---

### 第1回 組み込み開発の流れと環境(講義 90分)

---

組み込み開発環境を構成するソフトウェア、ハードウェア、ツールの概要と開発やデバッグの進め方を学ぶ。

#### (1) 組み込みシステムの特徴と開発環境

1. 組み込み機器開発の特徴
2. 組み込み OS
  - ・ 組み込み Linux の特徴(長所、短所)
  - ・ 開発のチェックポイント
3. 組み込み Linux 開発環境
  - ・ ホスト PC およびターゲットボード
  - ・ クロスコンパイル環境構築

#### (2) 組み込み Linux 実装

1. ブートローダ
2. ファイルシステム
3. Linux の実装
4. クロスコンパイル
5. 動作確認

#### (3) アプリケーション開発

1. ターゲットボードのリソース
2. システムコール
3. デバッグ手法
4. デバイスドライバ概要

#### (4) インタフェース利用プログラミング

1. ターゲットシステム要件定義
2. システム設計と開発環境の選択
3. アプリケーションプログラム作成
4. ROM 化

---

## 第2回 組み込み開発環境の概要(講義 90分)

---

組み込み開発環境の全体像とその構成、必要性を理解する。

### (1) 組み込み開発環境の必要要素

1. OS
  - ・ iTRON
  - ・ Linux
  - ・ OSEK
2. 言語
  - ・ ルールチェッカ
  - ・ コンパイラ／クロスコンパイラ
  - ・ エミュレータ
3. ミドルウェアドライバ
  - ・ 画像信号エミュレータ
  - ・ 音声信号エミュレータ
4. プログラミング
  - ・ フラッシュ開発キット
  - ・ PROM プログラマ
  - ・ オンボードプログラマ
5. エミュレータ
  - ・ オンチップデバッグ
  - ・ フルスペックエミュレータ

### (2) 統合開発環境の構築

1. プラットフォーム管理
2. エミュレーション管理
3. インタフェース管理
4. ドキュメント管理

---

### 第3回 組み込み開発環境を用いた開発手順(講義 90分)

---

開発環境を用いた開発方法の特徴と役割を理解する。

#### (1) 組み込み型コンピュータ応用システムの開発技術

1. 組み込み型コンピュータ応用システムの開発工程
  - ・ システム全体
  - ・ ハードウェア開発工程
  - ・ ソフトウェア開発工程
2. 必要な作業項目と開発環境に求められる機能
  - ・ システム設計段階
  - ・ ハードウェア設計段階
  - ・ ソフトウェア設計段階
  - ・ 検証段階

#### (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. 組み込みコンピュータ応用システム
2. 開発用マシンの準備
3. クロス開発環境

---

## 第4回 プログラムデバッグの環境(講義 90分)

---

組み込み開発の基本的なプログラムデバッグのためのツールや方法の特徴を理解する。

### (1) 基本的なデバッグの方法

1. printk()等基本的な命令を使ったデバッグ
2. GDBを使用したデバッグ
3. モニタプログラム

### (2) トレース機能

1. トレースの基本機能
  - ・ ターゲット実行データの記録
  - ・ 現象のシミュレート
  - ・ 秒単位のイベントの検知
  - ・ トレースメモリの評価
  - ・ データトレース可能
2. トレース機能の拡張
  - ・ 長時間トレース
  - ・ 分岐トレース
  - ・ ハードディスクドライブの利用
  - ・ トリガ設定可能(ポイント、エリア、シーケンシャル、カウントなど)
  - ・ タイムスタンプ機能

### (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

---

## 第5回 デバッガソフトを使用したデバッグ環境(講義 90分)

---

デバッガソフトを使用したデバッグ環境の全体像とその構成、必要性を理解する。開発環境を用いた開発方法の特徴と役割を理解する。

### (1) デバッガソフトによるカーネルやアプリケーションのデバッグ

1. デバッグ環境構築の留意点
  - ・ 接触不良の低減
  - ・ イーサネットは不要
  - ・ ターゲットにデバッグ用のポートを新たに設ける必要なし
  - ・ CPU クロックスピードの影響を受けない
  - ・ 量産ボードへの適用
2. フラッシュ ROM への対応
  - ・ サポートデバイス
  - ・ 簡単なメニュー操作でフラッシュ ROM に書き込み可能
  - ・ 書き込みシーケンスを意識せずにダウンロード

### (2) エミュレータの必要性

1. 各プロセスの状態遷移によるOS動作に関する検証
2. プロセスの状態遷移解析
3. プロセスのパターン分析
4. 異常処理時間のプロセス検索
5. OS アナライザ連動

---

## 第 6 回 ICE を使用したデバッグ環境(講義 90 分)

---

ICE を使用したデバッグ環境の全体像とその構成、必要性を理解する。開発環境を用いた開発方法の特徴と役割を理解する。

### (1) ハードウェアエミュレーションとは

1. CPU の特性
2. エミュレーションのための条件
3. エミュレーションする内容

### (2) ICE とは

1. ICE の目的
2. ハードウェアのエミュレーション方式
3. ハードウェア開発と並行したソフトウェア開発
4. Linux のエミュレータ環境

### (3) ICE によるハードウェア関連のデバッグ

1. リアルタイムトレース機能
2. リアルタイム実行時発生エラーの対処
3. 内部 I/O レジスタの設定や参照

### (4) Linux 対応 ICE (UniSTAC/J、UniSTAC II /J、UniSTAC/ASSP、EJ-Debug)

1. JTAG による接続
2. LAN、USB インタフェースによる接続

---

## 第7回 ツールチェーンによるデバッグ(講義 90分)

---

ツールチェーンの機能と特徴、利点/欠点、ツールチェーンを用いたデバッグの方法と手順を理解する。

### (1) ツールチェーンとは

1. ツールチェーンの構成
  - ・ コンパイラ
  - ・ binutils プログラム
  - ・ クロスコンパイラ機能
  - ・ ネイティブコンパイラ機能
2. ツールチェーンの要件
  - ・ ネイティブビッグエンディアン
  - ・ ネイティブリトルエンディアン
  - ・ ビッグエンディアンターゲット
  - ・ Linux/386 クロス環境
  - ・ リトルエンディアンターゲット

### (2) ツールチェーンの種類と特徴

1. make
2. GNU ツールチェーン
3. EABI ツールチェーン



---

## 第 8 回 組み込みアプリケーションデバッグの手順(講義 90 分)

---

組み込み開発時のデバッグ方法やトラブル解消の手法を理解する。

### (1) デバッグ対象と方法

1. 実機レベルのハードウェアの検証
2. Linux をターゲットマシンに組み込んだ後の統合デバッグ
3. デバッグするための接続
  - ・ LAN コネクタ
  - ・ シリアルデバッグ
  - ・ ブレークポイントの制限
  - ・ ソフトウェアブレーク無制限
  - ・ ハードウェアブレークが利用できる
  - ・ フラッシュ ROM 等のリードオンリーメモリでもブレーク
  - ・ メモリが正しく読み書きできているか
  - ・ バスチェック機能
  - ・ プログラムがリアルタイム動作で正常に動作しているか
  - ・ リアルタイムトレース機能

### (2) 割り込み発生時のトラブル解消

1. ドライバ開発時の割り込みと各 I/O との関連トラブル
2. 不足している資源(メモリ)の補完
  - ・ 不足した SRAM、SDRAM 等をエミュレーションメモリで代用
  - ・ ROM をエミュレーションメモリで代用
  - ・ システムの動作時間の計測
3. カバレッジパフォーマンスプロファイル測定機能

---

## 第9回 組み込みアプリケーションの事例と開発環境(ワークショップ 90分)

---

組み込みアプリケーションの開発事例と、それに対応する開発環境の構成、必要性を理解する。開発環境を用いた開発方法の特徴と役割を理解する。

### (1) 組み込みアプリケーション開発の事例

1. キャッシュとROM化
  - ・ 仕様
  - ・ 設計方式と必要な開発環境
2. マイコン周辺機能
  - ・ 仕様
  - ・ 設計方式と必要な開発環境
3. 周辺I/O制御プログラミング
  - ・ 仕様
  - ・ 設計方式と必要な開発環境
4. DRAM制御プログラミング
  - ・ 仕様
  - ・ 設計方式と必要な開発環境

### (2) クリティカルな処理の実現

1. スタートアップルーチン、例外処理、安全化
2. キャッシュを有効にした場合のROM化
3. 低消費電力モード、PCG、WDT、BSC
4. 設計方式と必要な開発環境

---

## 第 10 回 組み込みクロス開発環境の構築(講義 90 分)

---

組み込みクロス開発環境の全体像とその構成、必要性を理解する。開発環境を用いた開発方法の特徴と役割を理解する。

### (1) クロス開発環境とは

1. プロセッサとオペレーティングシステムの組み合わせ
2. アーキテクチャの違い
3. 異なるプロセッサ上で動作するプログラムの開発

### (2) クロス開発ツールの種類

1. 半導体ベンダが提供する開発環境
2. サードパーティが提供する開発環境
3. Free Software Foundation (FSF) が提供する GNU 開発環境

### (3) クロス開発環境の使用方法

1. ターゲット別の開発環境選択
2. 複数のターゲットに対応した環境
3. ライセンス構造

---

## 第 11 回 GNU 開発環境の特徴(講義 90 分)

---

GNU 開発環境の全体像とその構成、必要性を理解する。開発環境を用いた開発方法の特徴と役割を理解する。

### (1) GNU 開発環境とは

1. これまでの開発環境が抱える問題
2. GNU 開発環境の特徴
3. ライセンスとサポート
4. 操作体系

### (2) GNU 開発環境の構成

1. binutils (バイナリユーティリティ)
  - as
  - ld
  - nm
  - objdump
  - objcopy
  - strip
2. GDB (GNU デバッガ)
3. GCC (GNU コンパイラコレクション)
  - GNU C コンパイラ (gcc)
  - GNU C++ コンパイラ (g++)
4. 対応プロセッサ
5. 対応オペレーティングシステム

### (3) Newlib

1. 内容
2. できること
3. ライセンス

---

## 第 12 回 GNU 開発環境における組み込みアプリケーション開発(ワークショップ 90 分)

---

GNU 開発環境における組み込みアプリケーション開発を実際に行いながら理解する。

GNU 開発環境を用いた開発方法の特徴、役割、手順を理解する。

### (1) GNU 開発環境の入手

1. Cygwin 環境のダウンロードとインストール
  - ・ ソースコードの入手
  - ・ Cygwin のシェル環境
  - ・ バイナリコードのビルド
2. binutils, gcc, newlib, gdb 環境の構築
  - ・ 各パッケージの配布元
  - ・ ダウンロード
  - ・ Autoconf
  - ・ コンフィグレーション
  - ・ ビルド
  - ・ インストール

### (2) GDB スタブの ROM への書き込み

### (3) アプリケーションのコーディング、ビルド、デバッグ

---

## 第 13 回 組み込み Linux 開発最新デバッグ環境(講義 90 分)

---

組み込み Linux 開発最新デバッグ環境の全体像とその構成、必要性を理解する。開発環境を用いた開発方法の特徴と役割を理解する。

### (1) CPU 別の組み込み Linux ベンダ最新の対応状況

1. ARM7 系
2. ARM9 系
3. PXA250 系
4. PXA270 系
5. SH7750/51 系
6. FR-V 系

---

## 第 14 回 組み込み開発環境の評価(講義 90 分)

---

組み込み開発環境評価の観点とその内容、必要性を理解する。開発環境を用いた開発評価方法の特徴と役割を理解する。

- (1) 開発環境の評価
- (2) 使用目的の明確化
- (3) 開発環境の評価項目
  1. 購入および保守の費用
  2. 操作性と操作方法の習得期間
  3. 前提となる各種開発技法の習得の必要性
  4. ターゲットシステムのハードウェア仕様との適合性
  5. 他のツールとの接続性と生産物の互換性
- (4) 開発環境の開発効率および品質向上評価

---

## 第 15 回 組み込み開発環境におけるデバッグのパターン演習(ケースシヨップ 90 分)

---

組み込み開発環境におけるデバッグについて実際の事象ごとの手順と留意点を実習を行いながら理解する。

- (1) プログラムデバッグ演習
  1. 割り込み処理の不具合
    - ・ リアルタイム動作時に発生する異常を把握
    - ・ タイミングチャートの検討
  2. その他エラー処理のパターンとデバッグ方法
    - ・ メモリ保護エラー
    - ・ 表示装置(カラー液晶等)異常
    - ・ タイマーの不具合による暴走
    - ・ 通信データの異常値検出

以上