



To-Be エンジニア試験公式テキスト

情報Ⅱ

(システム構成・システム開発・ネットワークとマルチメディア)

To-Be エンジニア試験企画委員会 編著

コガク



目次

1章 システム構成 1~42

1 オペレーティングシステム (多くのシステムで搭載) 2

1.1 OSの種類 (Windows/Linux/ITRON など)	2
1.1.1 OSとは.....	2
1.1.2 OSの目的.....	3
1.1.3 パソコン用OS.....	3
1.1.4 組み込み用OS.....	3
1.2 OSの構成 (カーネル / ミドルウェアなど).....	4
1.2.1 カーネル	4
1.2.2 デバイス管理	5
1.2.3 共通ルーチン	5
1.2.4 ミドルウェア	6
1.3 記憶管理 (ヒープ, 仮想記憶など).....	7
1.3.1 メモリの構成	7
1.3.2 動的メモリと静的メモリ	7
1.3.3 メモリ保護.....	8
1.3.4 仮想記憶	9
1.4 タスク管理 (状態遷移, スケジューラなど).....	9
1.4.1 CPU 割り込み.....	10
1.4.2 マルチタスク	10
1.4.3 タスクの状態管理	11
1.4.4 スケジューリング	11

2 リアルタイムシステム (組み込み特有, ハードリアルタイム) 12

2.1 イベント (イベントドリブン).....	12
2.1.1 並列処理	12
2.1.2 割り込みの利用	13
2.1.3 イベント駆動	13
2.1.4 GUI 環境での応用	14

2.2 システムコール (トラップなど)	14
2.2.1 システムコールとは	14
2.2.2 システムコールの種類	15
2.2.3 カーネルモード	16
2.2.4 ソフト割込みとトラップ	16
2.3 優先度 (プリエンブションなど)	17
2.3.1 実時間制御と優先度	17
2.3.2 タスク管理ブロック	17
2.3.3 プリエンブション	18
2.3.4 多重割込み	19
2.4 割込みハンドラ (コンテキストなど)	19
2.4.1 割込みの仕組み	19
2.4.2 データの退避と復元	20
2.4.3 コンテキスト	20
2.4.4 デバッグでの応用	21



ソフトウェア部品 (部品として選択可能)

3.1 関数ライブラリ (開発効率向上)	22
3.1.1 ライブラリとは	22
3.1.2 共有ライブラリ (DLL)	23
3.1.3 C 言語のライブラリ	23
3.1.4 ソフトウェアコンポーネント	24
3.2 ミドルウェア (汎用機能の提供)	24
3.2.1 ミドルウェアとは	24
3.2.2 アプリケーションとプラットフォーム	25
3.2.3 ミドルウェアの分類	25
3.2.4 代表的なミドルウェア	26
3.3 カーネル (OS による機能提供)	26
3.3.1 カーネルとは	27
3.3.2 特権モード	27
3.3.3 リソース管理	28
3.3.4 カーネルの機能	28
3.4 デバイスドライバ (タスクと割込み)	29
3.4.1 デバイスドライバとは	29
3.4.2 ハードウェアと割込み	30
3.4.3 抽象化インターフェースの提供	30
3.4.4 標準ドライバ	31




ファイルシステム (外部記憶の利用)

4.1 ディレクトリ管理 (ルートディレクトリ)	32
4.1.1 ディレクトリとは	32
4.1.2 ツリー構造	33

4.1.3	ディレクトリの種類	33
4.1.4	ディレクトリによる管理	34
4.2	ファイル管理 (クラスタ, セクタなど)	34
4.2.1	ファイルとは	35
4.2.2	ファイルの分類	35
4.2.3	ファイルシステムの基本構造	36
4.2.4	セクタとクラスタ	36
4.3	FAT (アロケーションテーブル)	37
4.3.1	FAT とは	37
4.3.2	DOS のファイルシステム	38
4.3.3	ファイルフォーマット	38
4.3.4	ファイルシステムの進化	39
4.4	ファイル編成 (シーケンシャルアクセスなど)	39
4.4.1	ファイル編成法とは	39
4.4.2	メインフレーム	40
4.4.3	シーケンシャルファイル	40
4.4.4	その他のファイル編成法	41

2章 システム開発 43~86

	開発管理 (開発プロセス管理の基本)	44
1.1	PERT (日程計画)	44
1.1.1	開発管理の前提	44
1.1.2	PERT 図	45
1.1.3	クリティカルパス	45
1.1.4	日程の計算方法	45
1.2	従来型開発手法 (ウォーターフォールなど)	46
1.2.1	ウォーターフォールとは	47
1.2.2	成果物の明確化	47
1.2.3	進捗管理	48
1.2.4	ウォーターフォール型開発の問題点	48
1.3	反復型開発 (アジャイル, XP など)	49
1.3.1	反復型開発とは	49
1.3.2	反復型開発の適用	49
1.3.3	アジャイル	50
1.3.4	XP	50
1.4	レビュー (デザインレビュー, ウォークスルーなど)	51
1.4.1	レビューとは	51
1.4.2	レビューの進め方	52
1.4.3	デザインレビュー	52
1.4.4	ウォークスルー	53

2	開発ツール (開発の各工程で使用)	54
2.1	言語処理ツール (アセンブラ / コンパイラ / リンカなど)	54
2.1.1	エディタ	54
2.1.2	アセンブラ	55
2.1.3	コンパイラ	55
2.1.4	リンカ	56
2.2	IDE (統合開発環境)	56
2.2.1	統合開発環境の特徴	57
2.2.2	IDE の機能	57
2.2.3	ソフトウェア IDE	58
2.2.4	ハードウェア IDE	58
2.3	ICE (JTAG, インサーキットエミュレータ)	59
2.3.1	ICE とは	59
2.3.2	シミュレータとエミュレータ	60
2.3.3	ICE の機能	60
2.3.4	JTAG	61
2.4	デバッガ (ブレークポイントなど)	63
2.4.1	デバッガとは	63
2.4.2	シングルステップ	63
2.4.3	ブレークポイント	64
2.4.4	逆アセンブル	64

3	開発工程 (ソフトウェアの標準的開発工程)	65
3.1	要件定義 (分析 / 定義)	65
3.1.1	要件定義とは	65
3.1.2	要求定義と要件定義	66
3.1.3	要件定義の進め方	66
3.1.4	要件定義の成果物	67
3.2	設計 (方式 / 詳細)	67
3.2.1	ソフトウェア設計とは	68
3.2.2	方式設計と詳細設計	68
3.2.3	ソフトウェア設計の進め方	69
3.2.4	ソフトウェア設計の成果物	70
3.3	製造 (ソースコード)	70
3.3.1	ソフトウェア製造とは	70
3.3.2	ソフトウェア設計と製造	71
3.3.3	ソフトウェア製造の進め方	71
3.3.4	ソフトウェア製造の成果物	72
3.4	テスト (V 字モデル, 単体 / 結合)	72
3.4.1	ソフトウェアテストとは	73
3.4.2	V 字モデル	74
3.4.3	ソフトウェアテストの進め方	74

3.4.4	ソフトウェアテストの成果物	75
-------	---------------------	----

オブジェクト指向開発 (新しい考え方)

76

4.1	オブジェクト指向の考え方 (カプセル化など)	76
4.1.1	オブジェクト指向とは	76
4.1.2	カプセル化	77
4.1.3	継承	77
4.1.4	ポリモーフィズム	78
4.2	モデリング (表記法の統一)	78
4.2.1	モデリングとは	78
4.2.2	オブジェクト指向	79
4.2.3	モデル表記法の統一	79
4.2.4	モデル駆動型アーキテクチャ	80
4.3	UML (構造図, 振る舞い図)	81
4.3.1	UML とは	81
4.3.2	UML の活用	81
4.3.3	構造図	82
4.3.4	振る舞い図	83
4.4	リファクタリング (内部構造の再構築)	84
4.4.1	リファクタリングとは	84
4.4.2	リファクタリングとテスト	85
4.4.3	リファクタリングの方法	85
4.4.4	リファクタリングカタログ	86

3章 ネットワークとマルチメディア ... 87~130

ネットワークの基礎 (基本概念, 用語)

88

1.1	OSI モデル (物理層~アプリケーション層)	88
1.1.1	OSI モデルの位置づけ	88
1.1.2	OSI 下位層	89
1.1.3	OSI 上位層	89
1.1.4	OSI と TCP/IP	90
1.2	LAN 関連装置 (ハブ/ルータ/スイッチ/無線 LAN など)	90
1.2.1	ネットワーク関連装置	90
1.2.2	ハブ/スイッチ	91
1.2.3	ルータとゲートウェイ	92
1.2.4	無線 LAN	92
1.3	IP (IP アドレスなど)	93
1.3.1	IP の役割	93
1.3.2	IP アドレス	94

1.3.3	ネットマスク	95
1.3.4	特別なアドレス（ローカル）	95
1.4	TCP（ポート番号など）	96
1.4.1	TCPの役割	96
1.4.2	TCPとUDP	96
1.4.3	ポート番号の役割	97
1.4.4	特別なポート番号（Well-known port, NAT）	98



インターネット

（主なインターネット上アプリケーション）

	99
2.1	電子メール（SMTP/POP3など）	99
2.1.1	メール送信プロトコル	99
2.1.2	メール受信プロトコル	100
2.1.3	メールサーバ	101
2.1.4	メールの設定	102
2.2	Web（HTTP, URLなど）	102
2.2.1	ホームページのしくみ	102
2.2.2	マークアップ言語	103
2.2.3	HTTP	103
2.2.4	URL	104
2.3	ファイル転送（FTPなど）	105
2.3.1	ファイル転送プロトコル	105
2.3.2	クライアント / サーバ	106
2.3.3	FTP	106
2.3.4	アップロード / ダウンロード	107
2.4	ネット機器管理（SNMPなど）	108
2.4.1	ネットワーク管理の必要性	108
2.4.2	ネットワーク管理のしくみ	108
2.4.3	SNMP	109
2.4.4	MIB	109



マルチメディア

（高速デジタルネットワークに支えられた大容量通信）

	111
3.1	音声（PCM, サンプリング）	111
3.1.1	サンプリング定理	111
3.1.2	時間と周波数（帯域）	112
3.1.3	音声符号化の必要性	112
3.1.4	音声圧縮技術	113
3.2	静止画（JPEGなど）	113
3.2.1	静止画の符号化	113
3.2.2	可逆符号化と非可逆符号化	114
3.2.3	基本的な変換（DFT/DCT）	114
3.2.4	JPEG	115
3.3	動画（MPEGなど）	116

3.3.1	動画の符号化	116
3.3.2	フレーム間符号化	116
3.3.3	MPEG	117
3.3.4	MPEGの種類	117
3.4	圧縮 (ZIP, LZH など)	118
3.4.1	データ圧縮とは	118
3.4.2	ファイル圧縮	119
3.4.3	ZIP	119
3.4.4	LZH	120






セキュリティ (ネットワーク社会の大前提)

4.1	暗号化 (公開鍵 / 秘密鍵)	121
4.1.1	暗号化とは	121
4.1.2	暗号技術	122
4.1.3	秘密鍵暗号	122
4.1.4	公開鍵暗号	123
4.2	認証 (デジタル署名)	124
4.2.1	認証とは	124
4.2.2	デジタル署名とは	124
4.2.3	デジタル署名の仕組み	125
4.2.4	電子署名による認証	125
4.3	セキュリティ対策 (ウイルス, 脆弱性)	126
4.3.1	セキュリティとは	126
4.3.2	インターネット利用上の脅威	126
4.3.3	脆弱性とは	127
4.3.4	セキュリティ技術	128
4.4	コピーガード (著作権保護のしくみなど)	128
4.4.1	コピーガードとは	128
4.4.2	デジタル情報の特徴	129
4.4.3	著作権保護	129
4.4.4	CPRM	130

索引	131
-----------------	-----



1章 システム構成

情報 II	
	システム構成
	システム開発
	ネットワークと マルチメディア



オペレーティングシステム (多くのシステムで搭載)

1.1 OSの種類 (Windows/Linux/ITRONなど)

高度なコンピュータ応用システムでは、CPUを有効活用するために、OS（オペレーティングシステム, Operating System）を利用しています。

OSは「基本ソフト」とも呼ばれ、コンピュータシステムを利用しやすくする様々な機能を支えてくれる影の存在です。

パソコンやマイコン組み込みシステムにおけるOSの役割を理解しておきましょう。

1.1.1 OSとは

コンピュータシステムはCPUやハードウェアの構成がそれぞれ異なるため、これらを直接制御するプログラムを毎回開発すると開発効率や信頼性の問題が生じます。

OSはこの問題を解決するため、ハードウェアの違いを吸収した論理的インターフェース（API=Application Programming Interface）を標準化して、アプリケーションに提供します。

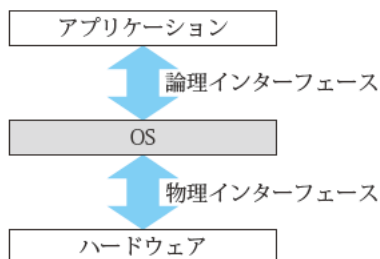


図 1.1 OSとは

1.1.2 OS の目的

OS の目的はシステムのオペレーション（操作、運用）のサポートです。これらを抽象化・標準化することで、アプリケーションの開発や他のシステムへの移植が容易になります。また、OS が一括して管理することで、資源を無駄なく活用することが可能になります。

OS の目的

- ・ **ハードウェアの抽象化**
アプリケーションの移植性、信頼性の実現。
- ・ 共通機能の提供
アプリケーション開発の容易化。
- ・ **資源（リソース）の管理**
複数タスクからのシステム資源利用の管理。

1.1.3 パソコン用 OS

パソコン用の OS として普及しているものとして、Microsoft 社の Windows や、Apple 社の MacOS, Linux などのいわゆる UNIX 系 OS があります。

基本的には OS が同じであれば同じアプリケーションが動作可能で、アプリケーションには対応している OS（のバージョン）が明示してあります。

主なパソコン用 OS

- ・ Windows
Microsoft 社のパソコン向け OS シリーズ
- ・ Mac OS
Apple 社のパソコン向け OS シリーズ
- ・ Linux
UNIX 系 OS。多くのプラットフォームに移植されている。

1.1.4 組み込み用 OS

組み込みシステム（Embedded System）はパソコンなどの人間相手のコンピュータ装置とは異なり、家電や産業用機械などに組み込まれるため、処理時間制限に対応できるリアルタイム OS であることが多く、ITRON や、VxWORKS などが有名です。

最近ではパソコンと携帯情報端末の境界があいまいで、Android が普及する一方、Linux・Windows の組み込み対応版も利用されています。

主な組み込み用 OS

- ITRON
TRON プロジェクトの産業用 OS
- VxWorks
Wind River 社の高信頼性 OS
- Android
Google 社の携帯情報端末向け OS

1.2 OSの構成（カーネル/ミドルウェアなど）

OSには多くの種類がありますが、基本的な構成は似ています。

システムを利用する場合には、通常ほとんど意識しない OS の一般的な構成を理解しておきましょう。

1.2.1 カーネル

カーネルとは OS の中心部分で、CPU システムのリソースを管理し、アプリケーションに API（Application Programming Interface）を提供します。

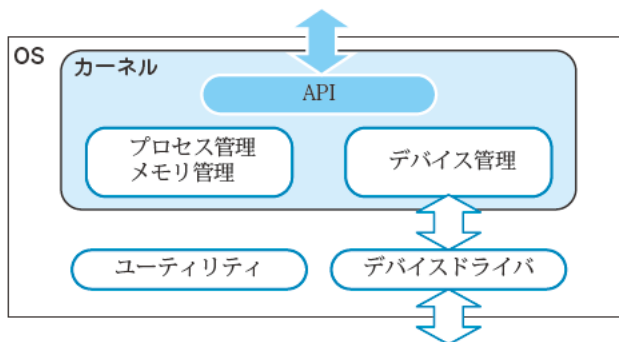


図 1.2 カーネル

アプリケーションプログラムとは異なり、CPU のシステム全体を制御するため、特別な権限を持つ動作モードが用意されています。

1.2.2 デバイス管理

システムに接続される入出力装置は一般に「デバイス」と呼ばれ、OS がその利用を共有資源として一括管理します。

特に複数のアプリケーションプログラムが同時に動作するマルチタスクシステムでは、他のプログラムの状態を無視して勝手にデバイスを使用するとシステム全体の問題を引き起こすため、デバイスの利用を確実に管理する必要があります。

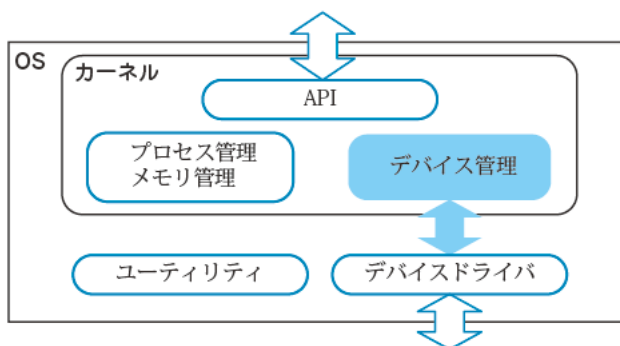


図 1.3 デバイス管理

1.2.3 共通ルーチン

パソコンなどではユーザの利便を図るため、エディタや通信用のツールなど、「ユーティリティ」と呼ばれる補助的機能を果たすプログラムが提供されます。

また、OS は「共通ルーチン」「ライブラリ」として開発の効率や信頼性を高めてくれる、サブルーチンや関数も提供します。