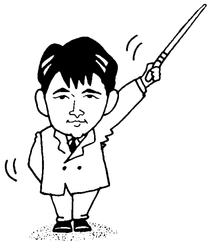


オペレーティングシステム

Operating System

－　２００５年度版　－

福岡工業大学
情報工学部 情報工学科
柴田望洋

BohYoh Shibata
Fukuoka Institute of Technology

本資料について

◆　本資料は、2005 年度・福岡工業大学 情報工学部 情報工学科２年生の講義

『オペレーティングシステム』

の補助テキストとして、福岡工業大学 情報工学部 情報工学科 柴田望洋が編んだものである。

◆　参考文献・引用文献等は、資料の最後にまとめて示す。

◆　諸君が本資料をファイルに綴じやすいように、研究室の学生達（卒研生と大学院生）が時間を割いて、わざわざ穴を開けるという作業を行っている（一度のパンチで開けることのできる枚数は限られており、**気の遠くなるような**時間がかかっている）。

必ずＢ５のバインダーを用意して、きちんと綴じていただきたい。

◆　本資料のプログラムを含むすべての内容は、著作権法上の保護を受けており、著作権者である柴田望洋の許諾を得ることなく、無断で複写・複製をすることは禁じられている。

本資料は、Microsoft 社のワープロソフトウェアである Microsoft Word 2003 を用いて作成した。

Copyright 2005 BohYoh Shibata

オペレーティングシステムとは

■ オペレーティングシステムと基本ソフトウェア

広義のオペレーティングシステム（*operating system*）すなわち基本ソフトウェア（*basic software*）は、主に以下の三つから構成されるソフトウェアです。

・制御プログラム（狭義のオペレーティングシステム）
・汎用言語プロセッサ
・サービスプログラム（ユーティリティプログラム）

● 制御プログラム（*control program*） ●

ジョブ管理、タスク管理、データ管理、入出力管理などを行い、コンピュータの資源を効率よく利用するためのソフトウェアです。

● 言語プロセッサ（*language processor*） ●

プログラムの翻訳などを行います。

● サービスプログラム（*service program*） ●

ファイルのコピー、ソート（整列）、連係編集プログラムなど、OS で提供されるユーティリティソフトウェアです。

第２種【平成７年度秋】午前・問 35

次の文は、言語プロセッサ、サービスプログラム及び制御プログラムについて記述したものである。ａ、ｂ、ｃの組合せとして正しいものはどれか。

a　資源割振り、スケジューリング、入出力制御、データ管理などを行う。
b　プログラムの翻訳などを行う。
c　整列プログラム、診断プログラム、追跡プログラムなど、一般的に必要とされるプログラムを提供する。

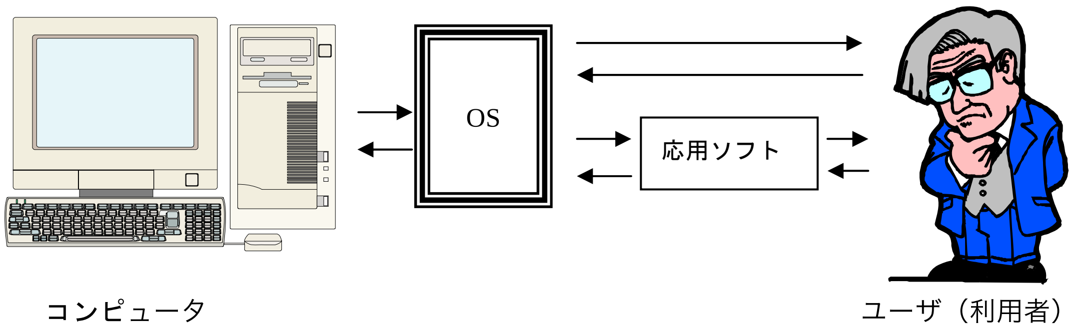
	a	b	c
ア	言語プロセッサ	サービスプログラム	制御プログラム
イ	サービスプログラム	言語プロセッサ	制御プログラム
ウ	サービスプログラム	制御プログラム	言語プロセッサ
エ	制御プログラム	言語プロセッサ	サービスプログラム
オ	制御プログラム	サービスプログラム	言語プロセッサ

Copyright 2005 BohYoh Shibata

■ オペレーティングシステム ■

コンピュータを利用する際には、（原則として）何らかのソフトウェアが動作している。そのソフトこそが、**オペレーティングシステム**（OS：*operating system*）である。

OSとは計算機を利用するために必要不可欠な一種の**基本的な**ソフトウェアであり、**リソース**（*resource*／資源）を管理し、よりよい環境を利用者に提供する。



◎ UNIX ◎

ワークステーション用の代表的なOSであり、マルチユーザ・マルチタスクをサポートします。

AT&T（*American Telephone & Telegraph*）社のベル研究所の研究員だったKen Thompsonが、ミニコンピュータDEC PDP-7を使いこなすために開発し始めたのがきっかけとなって作られたOSで1970年頃に完成しました。開発にはDennis Ritchieが加わり、C言語を産み出すことになりました。

彼らは、MITのMulticsプロジェクトに参加していたため、UNIXはMulticsの影響を大きく受けている（多くのことを目指すよりも、一つのことを目指したため、Multicsの［multi＝複］を［uni＝単］に置き換えたunicsに由来するネーミングが行われました）。

1975年、UNIXは大学などの研究所へ配布が開始され、さらに1978年には、Cコンパイラを含むようになり、世界中に普及することになりました。1983年には、Dennis RitchieとKen Tompsonは、UNIXオペレーティングシステムの設計と開発という功績により、チューリング賞を受賞しています。

技術的な特徴：

- ■ 入出力装置などのデバイスを、ファイルと同様に、ストリームとして扱うことができます。
- ■ いろいろなコマンドインタプリタ（文字によるコマンドによって対話的に処理を行うことができます）が提供されます。
- ■ ファイルを相対パス・絶対パスで指定することができます。
- ■ 仕様が公開されて移植性が高いので、広範囲の機種で利用できる。

Copyright 2005 BohYoh Shibata

◎ Microsoft Windows（3.1／95／98／Me／NT／2000／XP） ◎

パソコン用の代表的な OS。シングルユーザ・マルチタスクをサポートする。

現在、世界でもっとも有名なソフトウェア会社となっている Microsoft 社が開発（もともと某会社が作っていたものを買収）した OS である MS-DOS から発展した OS です。

いまや世界中に広がっており、その勢いは誰にも止められそうにありません。今後は、全ての家電製品にも MS-Windows 同様のルック＆フィールが採用されるとか（??）。

会長のビル・ゲイツ氏は、高校生の頃から全米的に有名になってしまった、（元）パソコン坊や。世界中のソフト業界が、この坊やの思いつきに左右されているといっても過言ではありません。この坊やの武勇伝などは、いろいろな伝書などで紹介されているので、ソフト業界を目指す人は、読んでおいても損はしません。

■ MS-Windows98 の前身である MS-Windows 3.1 は、MS-DOS という OS 上で動作するソフトウェアであり、OS そのものではありませんでした。また現在主流となっている MS-Windows XP は、Wiondows 95 や 98 ではなくて、Windows NT および Windows 2000 の後継です。

第２種【平成８年度春】午前・問 35

オペレーティングシステム（OS）の一つである UNIX の説明として、**適切でない**ものはどれか。

ア　キャラクタベースのコマンドを用いた対話方式のヒューマンインタフェースを提供する。
イ　仕様が公開されており移植性が高いので、広範囲な機種で採用されている。
ウ　シングルユーザ、マルチタスク OS である。
エ　分散処理を容易に実現するネットワーク機能を提供する。
オ　ワークステーションの代表的な OS である。

第２種【平成９年度春】午前・問 32

UNIX に関する記述として、正しいものはどれか。

ア　コマンドインタプリタは 1 種類に統一されている。
イ　シングルユーザ、マルチプロセスシステムである。
ウ　入出力装置をファイルと同等に扱うことができる。
エ　ファイルへのパスの表記は 1 通りである。

Copyright 2005 BohYoh Shibata

■ UNIXとC言語[2)] ■

1969年、大掛かりなMulticsプロジェクトは、失敗に終わろうとしていた。目標としていた、高速で使いやすいオンラインシステムはおろか、使えるだけというものすらまったくできあがっていなかったのだ。最後には、何とか動くMulticsを作り上げたのは確かだ。しかしIBMがOS/360で陥ったのと同じ罠に、彼らもやはり引っかかったのだ。彼らが作ろうとしたOSはあまりにも巨大で、それに引き換え、使えるハードウエアはあまりにも貧弱だった。Multicsは、解決しなくてはならない技術的問題の宝庫だっただけでなく、「小さいことは美しい」という教訓をCに残してくれたのだ。

幻滅と共にMulticsプロジェクトから手を引いたベル研の開発者たちは、次に取り組むべき課題を探していた。その一人Ken Thompsonは、別のOSを作成することに熱意を燃やし、何通もの提案書を上司に提出していた（片端からボツになったけど）。承認が出るのを待つ間に、Thompsonとその同僚のDennis Ritchieは、Thompsonが作った“Space Travel”というゲームソフトをほったらかしにされていたPDP-7というコンピュータに移植して楽しんでいた。Space Travelは太陽系を大雑把にシミュレートしており、グラフィック画面上で宇宙船を飛ばし、いろいろな惑星へ着陸することができた。そのためにThompsonは猛烈な努力をして、Multicsよりはるかに単純で軽快なOSの基礎をPDP-7に搭載したのだ。OSはすべてアセンブラで書かれていた。1970年、Brian KernighanはこのOSを“UNIX”と命名した。もちろんこれは、Multicsから学んだ教訓、何をしてはならないかをもじったものだ。

そう、卵と鶏のどちらが先かという点では、間違いなくUNIXがCよりも先に誕生したのである（UNIXの時間情報が、1970年1月1日からの通算秒で表されているのもこのためだ － ここからすべてが始まったのである）。アセンブラを使ったことは、困った問題も引き起こした。コードを書くのは骨の折れる仕事だし、デバッグや読むのにも手間がかかる。Thompsonは高級言語を使うメリットに惹かれていた。ただしMulticsプロジェクトで体験した、PL/Iの性能や複雑さはもう御免だった。Thompsonはちょっとの間FORTRANを試してみたがうまく行かず、Bという新しい言語を作成した。これは研究として作られたBCPLという言語を、PDP-7の8Kワードのメモリにインタプリタが載るように単純化したものだ。Bは決して成功を収めた言語ではない。ハードウエアのメモリ量の制約から、コンパイラではなくインタプリタしか用意できなかった。そのためUNIX本体を記述するには、Bの性能はあまりにも低すぎたのだ。

1970年になって、新しいマシンPDP-11への切り替えが行われると共に、型なし言語を開発に使うのには無理があることが明らかになった。性能的な問題もあり、ThompsonはPDP-11用のOSの開発にB言語を使わず、今回もアセンブラで書くことにした。Dennis RitchieはPDP-11の上で、型をサポートし、かつ高性能な“New B”を作ることにした。“New B”－名前はすぐに“C”に決まった － はインタープリタではなくコンパイラであり、型を導入し、変数を宣言してから使うという仕様になっていた。

■ オペレーティングシステムの主な働き

オペレーティングシステムの主要な機能を以下に示します。

（1）**ジョブ管理**（*job management*）

ジョブのスケジュール管理、ハードウェア資源の割当てなどを行う。

（2）**タスク管理**（*task management*）

タスクの実行を監視して、CPU の割当てや主記憶など、ジョブに与えられる資源の管理などを行う。

（3）**データ管理**（*data management*）

データを主記憶に割り付けたり、磁気ディスクへのファイルのアロケーション、ディレクトリの管理などを行う。

（4）**記憶管理**（*memory management*）

主記憶を効率よく活用するために、実記憶管理・仮想記憶管理などを行う。

（5）**運用管理**（*operation management*）

稼動状況の報告など運用業務を支援する。

（6）**障害管理**（*fault management*）

障害処理を行い、信頼性や安全性を向上させる。

（7）**入出力管理**（*input-output management*）

各入出力装置の制御を円滑に行う。

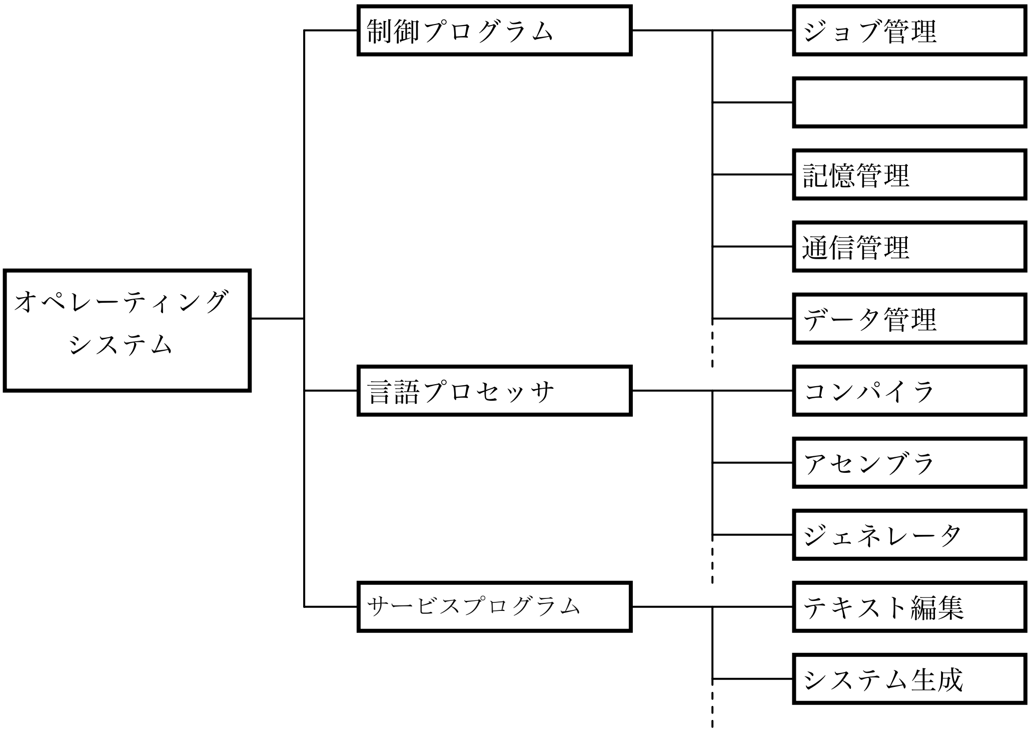
（8）**通信管理**（*communication management*）

複数のコンピュータ間の通信を制御する。

Copyright 2005 BohYoh Shibata

第２種【平成６年度秋】午前・問26

次の図は、オペレーティングシステムの機能の一部を体系化したものである。□に入れるべき適切な機能はどれか。



ア　オーバーレイ管理　　イ　カタログ管理　　ウ　タスク管理
エ　プロジェクト管理　　オ　メッセージ管理

【補足】

ここでの“オペレーティングシステム”は、広義のものであって、ここでの“制御プログラム”は、狭義のオペレーティングシステムです。

Copyright 2005 BohYoh Shibata

基本情報技術者試験について

就職活動をする際に、コンピュータ関連の資格をもっていると有利になる（少なくとも不利にならない）のは確かです。在学中にぜひ資格をとりましょう。

＊

私が大学院博士後期課程に在学していたとき、知人の紹介で公立の専門学校で講師のアルバイトをすることになりました。時給（50 分あたりの給料）は、次のようになっていました。

大学教授 … ¥5,500　　大学助教授 … ¥4,500　　大学生 … ¥2,500

しかし、給与規定は資格などによっても細分化されており、第２種情報処理技術者（現在の基本情報技術者）取得者の時給は¥3,500 で、私がもっていた資格である第１種情報処理技術者（現在のソフトウェア開発技術者）取得者は¥4,500 でした！

合格に対する一時金や、資格に対する手当て（毎月の給料に¥3,000〜¥9,000 がプラスされる！）が支給される会社もあります。

＊

基本情報技術者試験は、春と秋、年２回行われます。詳しくは、以下のホームページを参照してみてください。

http://www.jitec.jp/

Copyright 2005 BohYoh Shibata

OS とミドルウェア

ミドルウェア

基本ソフトウェアと応用ソフトウェアの間に位置して、多数の利用者が共通に使用する基本処理機能を提供するソフトウェアのことです。

異なるメーカの機種間での応用ソフトウェアの移植性の確保や、相互接続を容易にするために導入されます。代表的なものとして、

・データベース管理システム
・通信管理システム
・ソフトウェア開発支援ツール
・日本語ワードプロセッサ
・表計算ソフト
・グラフィック処理システム

などがあります。

ソフトウェアの分類を以下に示します。

ソフトウェア
- ■システムソフトウェア
  - □基本ソフトウェア
  - □ミドルウェア
- ■応用ソフトウェア
  - □個別応用ソフトウェア
  - □共通応用ソフトウェア

第２種【平成 10 年度秋】午前・問 43

異なるメーカの機種間での応用ソフトウェアの移植性の確保や、相互接続を容易にするために導入されるソフトウェアであり、システムの構成を次のように 4 階層で表現した場合に、[　　]に位置付けられるものはどれか。

応用ソフトウェア
a
基本ソフトウェア
ハードウェア

ア　グループウェア　　イ　シェアウェア
ウ　ファームウェア　　エ　ミドルウェア

Copyright 2005 BohYoh Shibata

情報処理システムの基本

OS と関連深い事項として、情報処理システムについて簡単に学習しましょう。情報処理システムは、以下のように発展してきました。

（1）バッチ処理からオンライン処理へ

（2）集中処理→分散処理→エンドユーザコンピューティング

（3）定量的情報処理→定性的情報処理（マルチメディア化を含む）

■ バッチ処理方式（*batch processing*） ■

> ユーザからの仕事を蓄えておき、都合の良いときに（あるいは一定の周期で）逐次処理を行っていく方式。

ジョブ（*job*）

プログラム実行の一連の作業

入力		処理		出力
カードリーダ （キーボード）	→	**プログラム**	→	**プリンタ** （ディスプレイ）

バッチ処理方式では、ジョブをまとめて逐次処理する。

A → B → C

利用者は、プログラムやデータなどをセンターに提出し、センターでは、一定期間分のジョブを一括処理する。ジョブを実行開始することを、「ジョブを投入する」といい、その開始・実行などを制御する言語を**ジョブ制御言語**（JCL：*job control language*）と呼ぶ。

特徴

（1）　オペレーティングシステムの働きにより、ジョブの切換え作業を自動化できるので、コンピュータの利用効率を上げることができる。

（2）　ユーザに結果が戻るまでの**応答時間＝ターンアラウンドタイム**が長い。

センターバッチ処理（*center batch processing*）

↓

リモートバッチ処理（*remote batch processing*）

入出力装置をユーザの近く（センターにとっては遠隔地）に設置して、ターンアラウンドタイムを縮小

Copyright 2005 BohYoh Shibata

（３）　入出力には、時間がかかるため、その間に CPU が遊んでしまう。これをアイドル時間と呼びます。

→　アイドルの意味は、英和辞典で調べてみましょう。

【例】給与計算、売り上げ集計（日次、月次、年次）、成績集計、
大規模な科学技術計算…

第２種【平成７年度春】午前・問 55

次の条件を満たすコンピュータの処理形態として、最も適切なものはどれか。

・入出力はコンピュータの設置場所と離れた遠隔装置から行う。
・あらかじめ蓄えられたデータをまとめて処理する。

ア　オンラインリアルタイム処理　イ　対話型処理　ウ　バッチ処理
エ　リモートバッチ処理　オ　ロールバック処理

第２種【平成 12 年度秋】午前・問 33

あるジョブのターンアラウンドタイムを解析したところ、1,350 秒のうち CPU 時間が 2／3 であり、残りは入出力時間であった。１年後に、CPU 時間はデータ量の増加を考慮しても、性能改善によって当年比 80％に、入出力時間はデータ量の増加によって当年比 120％になることが予想されるとき、このジョブのターンアラウンドタイムは何秒になるか。ここで、待ち時間、オーバヘッドなどは無視する。

ア　1,095　イ　1,260　ウ　1,500　エ　1,665

Copyright 2005 BohYoh Shibata

■ オンラインシステム（*on-line system*） ■

ユーザからの要求（データの発生等）に対して即座に処理を行う方式。

遠隔地において、要求の発生と同時に処理をする方式。**リアルタイムシステム**（*real-time system*）。**応答時間＝レスポンスタイム**が短いことが要求される。

特徴

（１）要求の発生に対して即座に処理が行われ、応答時間が短い。
（２)座席の予約状況、口座の預金状況などの必要な情報は常に更新されている。
　※そのために、システムが複雑・高価になりやすい。
（３）システムの運営、とくに信頼性、保安性などの維持などが大変である。

【例】　座席予約システム、
　銀行のオンラインシステムＡＴＭ（*Automatic Teller Machine*:自動預金機）
　プロセス制御、交通管制システムなど

二つの応答時間

バッチ処理方式の応答時間 → ターンアラウンドタイム
オンライン処理方式の応答時間 → レスポンスタイム

ターンアラウンドタイム（*turnaround time*）

直訳すると往復所要時間。ジョブを投入してから、最終的な処理結果が出力されるまでの時間。

レスポンスタイム（*response time*）

利用者が問合せ又は要求終了を端末で指示してから、その端末に処理結果の出力が始まるまでの時間。たとえば、ゲームにおいて、キャラクタを移動するキーボード操作に対して、すぐにそのキャラクタは移動すべきです。もしレスポンスタイムが長ければ、キーボードを操作してから、キャラクタが移動するまでにタイムラグが発生し、使いにくいものとなってしまいます。

Copyright 2005 BohYoh Shibata

第２種【平成 6 年度秋】午前・問 65

オンラインシステムにおいて、利用者が問合せ又は要求終了を端末で指示してから、その端末に処理結果の出力が始まるまでの時間を何というか。

ア　アクセス時間　　イ　応答時間　　ウ　ターンアラウンドタイム
エ　プロセス時間　　オ　リアルタイム

第２種【平成 7 年度秋】午前・問 52

コンピュータシステムに対して問合せ又は要求の終わりを指示してから、利用者端末に最初の応答が出始めるまでの時間を何というか。

ア　アクセスタイム　　イ　サイクルタイム　　ウ　ターンアラウンドタイム
エ　リアルタイム　　オ　レスポンスタイム

第２種【平成 9 年度春】午前・問 52

オンライン検索システムにおいて、利用者が検索コマンドの送信を端末で指示してから、その端末に検索結果の最初の出力が始まるまでの時間を何というか。

ア　アクセス時間　　イ　位置決め時間　　ウ　応答時間　　エ　ターンアラウンドタイム

第２種【平成 11 年度秋】午前・問 52

コンピュータシステムに対して問合せ又は要求の終わりを指示してから、利用者端末に最初の処理結果のメッセージが出始めるまでの時間を何というか。

ア　アクセスタイム　　イ　サイクルタイム
ウ　ターンアラウンドタイム　　エ　レスポンスタイム

Copyright 2005 BohYoh Shibata

■ TSS＝タイムシェアリングシステム（time-sharing system）　■

短い時間間隔で、ユーザを切換えてサービスを行う方式。

オンラインシステムでは、データ等は遠隔地から送ることができるが、実際のジョブはセンターで行われる。

↓

ジョブ自身を、利用者が行いたい。すなわち、複数の人間が端末において、あたかも自分が計算機を占有しているかのように使いたい。

コンピュータが、短い時間間隔で利用者を切り換えて**会話的な**（*interactive*）サービスを行う方式。

特徴

（１）対話型処理が可能。
（２）ユーザはコンピュータを占有しているかのようにプログラムの実行が可能。
（３）コンピュータの持つ機能を共同利用できる。

【例】　大学、共同利用施設、文献検索など

Copyright 2005 BohYoh Shibata

■ 分散処理方式（distributed processing）

> 複数のコンピュータが対等に結合し、情報のやり取りを行うコンピュータネットワーク上において、複数のコンピュータを利用する方式

複数のコンピュータを対等に結合し、互いに情報のやり取りを行うコンピュータネットワークにおいて、複数のコンピュータを利用する方式。

コンピュータネットワーク：

ＬＡＮ（*local area network*）

ＷＡＮ（*wide area network*）

分散処理において、処理の大部分は自分の部門のコンピュータで行い、自分のコンピュータで処理しきれないものを、ネットワークを通じて、より高速・大型のコンピュータや、機器などを利用する。

集中処理システムと分散処理システムの長所と短所を比較すると、次のようになります。

	集中処理システム	分散処理システム
長所	■センタに集中して対策を施すことによって、機密保護やセキュリティやデータの一貫性を維持・管理することが容易である。 ■ネットワークの性能はシステム全体の性能にあまり影響を与えない。 ■管理の集中化によって、人件費を節約できる。 ■コストパフォーマンスが低くなる傾向がある。	■個々のシステムの資源管理が容易である。 ■災害や障害の際の損害が局所化される。 ■機能や設備が分散しているのでシステムの変更が容易。 ■安価なシステムを組み合わせることにより、経済的なシステムの構築が容易である。
短所	■一部の装置の故障がシステム全体の停止につながることが多く、災害や障害の際の修復作業は容易ではない。 ■機能の拡張や業務量の増大に対応したシステムの拡張などが困難である。 ■システムの規模が大きくなりやすい。 ■規模が大きくなると、こまわりがきかないため新技術の導入が困難になりやすい。	■システム全体を効率よく運用するための運用管理が複雑になりやすい。 ■異常が発生した際に、原因の特定が困難である。 ■ネットワークにおけるデータの遅延が全体の性能を低下させる。 ■情報の安全管理、末端での管理が困難である。

Copyright 2005 BohYoh Shibata

第２種【平成７年度秋】午前・問 54

集中処理システムと比較した場合の分散処理システムの特徴に関して、正しい記述はどれか。

ア　一部の装置の故障がシステム全体の停止につながることが多い。

イ　機能の拡張や業務量の増大に対応したシステムの拡張などが困難である。

ウ　機密保護やセキュリティの確保が容易である。

エ　システム全体を効率よく運用するための運用管理が複雑になりやすい。

オ　ネットワークの性能はシステム全体の性能にあまり影響を与えない。

第２種【平成９年度秋】午前・問 54

集中処理システムと比較した場合の分散処理システムの特徴に関して、正しい記述はどれか。

ア　一部の装置の故障がシステム全体の停止につながることが多い。

イ　機能の拡張や業務量の増大に対応したシステムの拡張などが困難である。

ウ　機密保護やセキュリティの確保が容易である。

エ　システム全体を効率よく運用するための運用管理が複雑になりやすい。

第２種【平成 11 年度秋】午前・問 55

分散処理システムと比較したとき、集中処理システムの記述として、最も適切なものはどれか。

ア　災害や障害のときにセンタ側で集中した修復作業を行うことができるので、システム全体が長時間停止する危険性を回避できる。

イ　システムを一括管理しているので、システム機能の追加・変更などの要求に応ずることが容易であり、バックログの堆積が起こりにくい。

ウ　センタに集中して対策を施すことによって、セキュリティやデータの一貫性を維持・管理することが容易である。

エ　ハードウェア及びソフトウェア資源の運用・管理が煩雑になるが、新技術に対応した拡張が容易である。

基本【平成 13 年度春】午前・問 37

広範な地域に配置した複数の計算機システムで構成される分散処理システムと単一のセンタで運用される集中処理システムを比較したとき、集中処理システムの特徴として、最も適切なものはどれか。

ア　災害や障害のときにセンタ側で集中した修復作業を行うことができるので、システム全体が長時間停止する危険性を回避できる。

イ　システムを一括管理しているので、システム機能の追加・変更などの要求に応ずることが容易であり，バックログの堆積が起こりにくい。

ウ　センタに集中して対策を施すことによって、データの一貫性を維持・管理しやすい。

エ　ハードウェア及びソフトウェア資源の運用・管理が煩雑になるが、新技術に対応した拡張が容易である。

Copyright 2005 BohYoh Shibata

■ 演習 ■

１．OS の正式名称を英語で書け。 ____________

２．空欄を埋めよ。

OS は、一種の［(a)　　　］なソフトウェアであり、［(b)　　　］を管理して、よりよい環境を利用者に提供する。　(a)________ (b)________

３．○×で答えよ。

(a)バッチ処理方式は、会話的な処理が可能である。 ______
(b)バッチ処理方式は、ターンアランドタイムが短くなる傾向がある。 ______
(c)バッチ処理方式では、ユーザは手元のコンピュータで処理ができる。 ______
(d)銀行の ATM に適しているのはオンラインシステムである。 ______
(e)オンラインシステムでは、ユーザの要求に対して即座に処理が行われる。 ______
(f)オンラインシステムでは、レスポンスタイムが長くなる傾向がある。 ______
(g)オンラインシステムは、システムの運営保守が容易である。 ______
(h)TSS では、対話型処理を行うことはできない。 ______
(i)TSS では、ユーザが 1 台のコンピュータを占有しているかのように利用できる。 ______
(j)飛行機の座席予約システムに適しているのは分散処理方式である。 ______
(k)給与計算に適しているのはオンラインシステムである。 ______

４．以下の語句の正式名称を英語で示せ。

(a)ＪＣＬ　(b)ＴＳＳ　(b)ＬＡＮ　(c)ＷＡＮ

(a)________ (b)________ (c)________ (d)________

学籍番号				氏名		
1						
2	(a)			(b)		
3	(a)		(b)		(c)	
	(d)		(e)		(f)	
	(g)		(h)		(i)	
	(j)		(k)			
4	(a)					
	(b)					
	(c)					
	(d)					

Copyright 2005 BohYoh Shibata

タスク管理

■ コンピュータ内部での処理形態 ■

１．ユニプロセッサ・ユニプログラミング方式（*uniprocessor-uniprogramming*）

１台の CPU を有するコンピュータが、１個のプログラムを処理する。

２．マルチ（多重）プログラミング方式（*multiprogramming*）

主記憶上に複数のプログラムを格納し、プログラムを切り替えることによって、コンピュータの利用効率を向上させる方式。最近はパソコンでもサポートされている。

※実際に同時に動作するプログラムは１個。

３．マルチプロセッサ方式（*multiprocessor*）

複数の CPU を用いて、マルチプログラミングをサポートする方式。

※CPU の数のプログラムが同時に動作。

Copyright 2005 BohYoh Shibata

■ タスク管理の目的

タスクとは、オペレーティングシステムから見たときの、ジョブの制御単位のことです。

タスク管理の主目的は、**CPU を効率よく利用する**ことです。

現在のコンピュータでは、主記憶上に複数のプログラムを格納しておき、プログラムを切りかえることによって、コンピュータの利用効率を向上させる方式である**マルチプログラミング**（**多重プログラミング**あるいは**マルチタスキング**とも呼ばれる）が行われます。

そのようなシステムでは、CPU が、あるプログラムから他のプログラムへと非常に短い時間で切り替えられ実行される。ある瞬間には単一のプログラムのみが実行されているのですが（CPU が 1 個の場合）、秒単位でみると、複数のプログラムが同時に実行されているかのように見えます。

■ プロセス（*process*）

コンピュータのユーザは、計算機の中に入り込むことはできません。OS の初期の時代では、"実行中のプログラム" をプロセスと考えてきました。現在では、**リソースを要求するユーザの代理**であると考えられています。プロセスはリソースを要求し、リソースはプロセスによって要求されます。

■ 仮想ＣＰＵ（virtual CPU）

各プロセスには、あたかも自分占有の CPU を有して実行しているかのように見えますが、このように各プロセスに対して作られた見せかけの CPU を**仮想 CPU**（*virtual CPU*）と呼びます。プロセスは、CPU の処理時間の一部分しかサービスを受けられないため、仮想 CPU の実行速度は、CPU より遅くなることになります。

■ ＣＰＵスケジューリング（*CPU scheduling*）

一般に CPU の数は、全ての実行中のプログラム（プロセス）に割り当てることができるほど十分ではなく、プログラムの間で切り換えて利用する必要がある。

このような切り換えを行うことを **CPU スケジューリング**と呼び、それを行うプログラムを **CPU スケジューラ**（*CPU scheduler*）あるいは**ディスパッチャ**（*dispatcher*）と呼ぶ。

Copyright 2005 BohYoh Shibata

第２種【平成７年度春】午前･問 33

次に示す機能は、汎用計算機のオペレーティングシステムにおけるジョブ管理、タスク管理、データ管理及び入出力管理のいずれかに含まれる機能である。

この中で、タスク管理に含まれるものはどれか。

ア　ＣＰＵ割当て　　イ　ジョブスケジュール機能　　ウ　スプール機能

エ　入出力の実行機能　　オ　ファイルの保護機能

第２種【平成 10 年度春】午前･問 32

オペレーティングシステムのタスク管理に含まれる機能はどれか。

ア　CPU 割当　　イ　スプール制御　　ウ　入出力の実行　　エ　ファイル保護

第２種【平成８年度秋】午前･問 32

タスク管理の目的として適切なものはどれか。

ア　CPU を効率よく使用する。
イ　運用管理者によって行われる運用管理業務を支援する。
ウ　記憶領域を効果的に使用するとともに、主記憶の容量の制約を緩和する。
エ　ハードウェアを意識しないデータの蓄積、処理、保存、運用を可能にする。
オ　ハードウェアを監視し、運用状況を把握してシステムの維持管理を行う。

第２種【平成９年度秋】午前･問 35

オペレーティングシステムのタスク管理の役割として、正しいものはどれか。

ア　各種の補助記憶装置へのアクセス手段を、装置に依存しない形態で提供し、応用プログラム作成の負担を軽減する。
イ　仮想記憶空間を提供し、実記憶を有効に利用する。
ウ　入出力装置の制御を行い、正確かつ効率よく入出カ装置を動作させる。
エ　マルチプログラミングの制御を行い、CPU を有効に利用する。

第２種【平成 11 年度秋】午前･問 32

OS におけるタスク管理の目的はどれか。

ア　オペレータにコマンド形式の対話インタフェースを提供する。
イ　仮想記憶を効率的に実現する。
ウ　処理装置の利用効率を高めるように制御する。
エ　ハードウェアを意識しないで、データが処理できるようにする。

Copyright 2005 BohYoh Shibata

第２種【平成12年度春】午前･問34

タスク管理の役割として、適切なものはどれか。

ア　各種の補助記憶装置へのアクセス手段を、装置に依存しない形態で提供し、応用プログラム作成の負担を軽減する。
イ　仮想記憶空間を提供し、実記憶を有効に利用する。
ウ　入出力装置の制御を行い、正確かつ効率良く入出力装置を動作させる。
エ　マルチプログラミングの制御を行い、CPU を有効に利用する。

第２種【平成６年度秋】午前･問62

オペレーティングシステムにおいて、あるプログラムの実行中に、入出力など処理装置に待ちが生じたとき、処理装置を直ちに別のプログラムの実行に割り当てる。このようにして、システム全体としての処理効率の向上をねらう方式を何というか。

ア　エミュレーション　　イ　タイムシェアリング　　ウ　多重プログラミング
エ　パイプライン処理　　オ　リアルタイム処理

第２種【平成８年度秋】午前･問33

オペレーティングシステムが、CPU 資源を割り当てる対象を示す用語を二つ選べ。

ア　クライアント　　イ　セグメント　　ウ　タスク　　エ　プロセス　　オ　ユーザ

第２種【平成８年度春】午前･問31

あるプログラムの実行中に、入出力などのために処理装置が待ち状態になったとき、処理装置を他のプログラムの実行に割り当てることによって処理装置を有効に利用する方式を何というか。

ア　オーバレイ　　イ　スラッシング　　ウ　ダイナミックアロケーション
エ　マルチプログラミング　　オ　ラウンドロビン

第２種【平成９年度春】午前･問51

オペレーティングシステムにおいて、あるプログラムを実行中に、入出力などで CPU に待ちが生じたとき、CPU を直ちに別のプログラムに割り当てる。

このような機能を実現するために、オペレーティングシステムがもつべき処理機能を何というか。

ア　タイムシェアリング　　イ　マルチタスキング
ウ　マルチプロセッシング　　エ　リアルタイム処理

Copyright 2005 BohYoh Shibata

第2種【平成10年度秋】午前･問35

オペレーティングシステムにおいて、あるプログラムを実行中に、入出力などで CPU に待ちが生じたとき、CPU を直ちに別のプログラムに割り当てる。

このような機能を実現するために、オペレーティングシステムがもつべき処理機能を何というか。

ア　タイムシェアリング　　イ　マルチタスキング
ウ　マルチプロセッシング　　エ　リアルタイム処理

基本【平成13年度春】午前･問32

あるプログラムの実行中に、入出力などのために処理装置が待ち状態になったとき、処理装置をほかのプログラムの実行に割り当てることによって有効に利用する方式を何というか。

ア　スラッシング　　イ　ダイナミックアロケーション
ウ　マルチプログラミング　　エ　ラウンドロビン

第2種【平成11年度秋】午前･問34

多重プログラミングに関する記述として、適切なものはどれか。

ア　ジョブとしては多重で処理されるが、シングルタスクで実行する。
イ　タスクの実行中に、入出力などを行ったために生じる CPU の空き時間を利用して、別タスクを並列に実行する。
ウ　プログラムの実行中に、自分自身を呼び出して実行するプログラミング方法である。
エ　プログラムを並列に処理するので、ハードウェアとして複数のプロセッサとメモリを結合した並列処理システムが必要である。

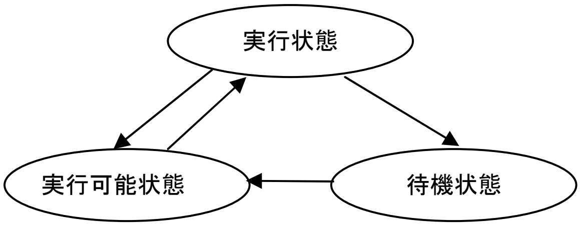
基本【平成13年度春】午前･問33

OS の機能の中で、実行可能状態にあるプロセスの中から次に実行すべきプロセスに CPU の使用権を与えて、実行状態にするものはどれか。

ア　アロケータ　　イ　イニシエータ　　ウ　ターミネータ　　エ　ディスパッチャ

Copyright 2005 BohYoh Shibata

■ プロセスの状態遷移



実行（running）

CPU を確保し、プログラムを実行している状態。

実行可能（*ready*）

CPU が割り当てられればすぐプログラムを実行できる状態。

待機（*wait*）

要求しているリソース、たとえば入出力機器が、空くのを待っているとか、メッセージが到達するのを待っているとか､何らかの待ち状態にあり､CPU を必要としていない状態。

▲ 実行可能状態から実行状態へ

CPU スケジューラ（*CPU scheduler*）すなわちディスパッチャによって、CPU を割り当てられたとき。

▲ 実行状態から実行可能状態へ

プリエンプション（より優先度の高いジョブに割り込まれることによって実行が中断される）によって、あるいは自分に割り当てられた時間＝**クオンタム**（*quantum*）を使い果たしたときに生じる。

▲ 実行状態から待ち状態へ

プロセスが入出力動作を起動し、その完了を待たなくなければいけなくなったとき、何らかのリソースが必要となり､それが空くのを待たなければならなくなったときに生じる。

他の遷移と異なり、プロセス自らの操作によって生じる。

▲ 待機状態から実行可能状態へ

入出力動作が完了したり、他のプロセスの命令により待ち状態が解除されたとき。

Copyright 2005 BohYoh Shibata

第２種【平成８年度春】午前・問 32

タスクの状態に関する次の記述中の[　　]に入れるべき適切な字句の組合せはどれか。

タスクは三つの状態で管理される。これらは、CPU 使用権を与えられた[a]状態、CPU 使用権を持っている[b]状態、及び入出力要求などをきっかけに[a]状態から移行する[c]状態である。

	a	b	c
ア	受付け	待ち	実行
イ	実行	実行可能	待ち
ウ	実行	待ち	割込み
エ	実行可能	受付け	割込み
オ	実行可能	割込み	実行

第２種【平成 12 年度春】午前・問 33

プロセスは、実行可能状態（ready）、実行状態（running）、待ち状態（wait）を遷移しながら実行される。プロセスの状態遷移に関する記述のうち、適切なものはどれか。

ア　CPU 処理と入出力処理が交互に現れるプロセスを複数個同時に実行させると、各プロセスは実行状態と待ち状態の二つの状態間だけを遷移する。
イ　実行可能状態とは、CPU の割当てを待っている状態をいう。実行可能状態のプロセスは一般に複数個存在し、これらは待ち行列を形成する。
ウ　時分割処理を行っているシステムでは、実行状態のプロセスは、一定時間が経過すると、待ち状態に遷移する。
エ　マルチプログラミングシステムでは、CPU が 1 個でも、実行状態のプロセスは複数個存在する。

第１種【平成８年度】午前・問 23

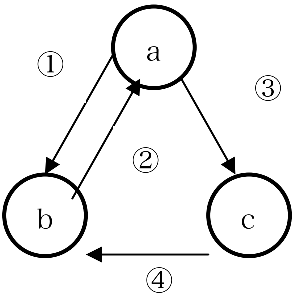
タスクは、実行可能状態（ready）、実行状態（running）、待ち状態（wait）の三つの状態を繰り返しながら実行される。タスクの状態遷移に関する記述のうち、**誤っているもの**はどれか。

ア　実行可能状態におかれたタスクが、待ち状態に移されることはない。
イ　実行状態のタスクは、ある種の事象の完了（例えば入出力動作の完了）を待つ必要が生じたとき、待ち状態におかれる。
ウ　実行状態のタスクは、自分に割り当てられた CPU 時間が終了すると待ち状態におかれる。
エ　タスクの実行が完了すると、そのタスクが使用していた資源は解放され、他のタスクが使用できるようになる。
オ　待ち状態のタスクは、待ちの原因となった事象が完了すると、実行可能状態に移る。

Copyright 2005 BohYoh Shibata

第2種【平成7年度秋】午前・問39

図はプロセスの状態と遷移を表している。a、b、c の状態を表しているのはどれか。

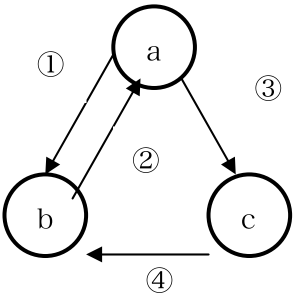


状態遷移の要因
①実行優先度の高いプロセスに CPU 使用権が移された。
②CPU 使用権が与えられた。
③入出力動作の完了を待つ。
④入出力動作が完了した。

	a	b	c
ア	実行可能状態	実行状態	待機状態
イ	実行可能状態	実行状態	実行状態
ウ	実行状態	実行可能状態	待機状態
エ	実行状態	待機状態	実行可能状態
オ	待機状態	実行状態	実行可能状態

第2種【平成11年度秋】午前・問33

図はプロセスの状態と遷移を表している。a、b、c の状態を表しているのはどれか。



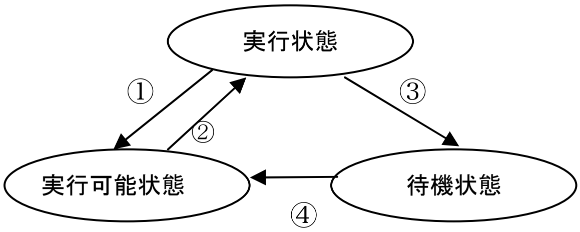
状態遷移の要因
①実行優先度の高いプロセスに CPU 使用権が移された。
②CPU 使用権が与えられた。
③入出力動作の完了を待つ。
④入出力動作が完了した。

	a	b	c
ア	実行可能状態	実行状態	待機状態
イ	実行可能状態	実行状態	実行状態
ウ	実行状態	実行可能状態	待機状態
エ	実行状態	待機状態	実行可能状態

Copyright 2005 BohYoh Shibata

第２種【平成９年度秋】午前・問 38

図はプロセスの状態と遷移を表している。状態遷移の要因①～④の正しい組み合せはどれか。



	①	②	③	④
ア	CPU 使用権が与えられた。	実行優先度の高いプロセスに CPU 使用権が移された。	入出力動作が完了した。	入出力動作の完了を待つ。
イ	実行優先度の高いプロセスに CPU 使用権が移された。	CPU 使用権が与えられた。	入出力動作の完了を待つ。	入出力動作が完了した。
ウ	入出力動作が完了した。	入出力動作の完了を待つ。	CPU 使用権が与えられた。	実行優先度の高いプロセスに CPU 使用権が移された。
エ	入出力動作の完了を待つ。	入出力動作が完了した。	実行優先度の高いプロセスに CPU 使用権が移された。	CPU 使用権が与えられた。

第１種【平成７年度】午前・問 27

図はオペレーティングシステムのタスクの状態遷移を表す。a，b，c の各状態に当てはまる組合せとして、正しいものはどれか。

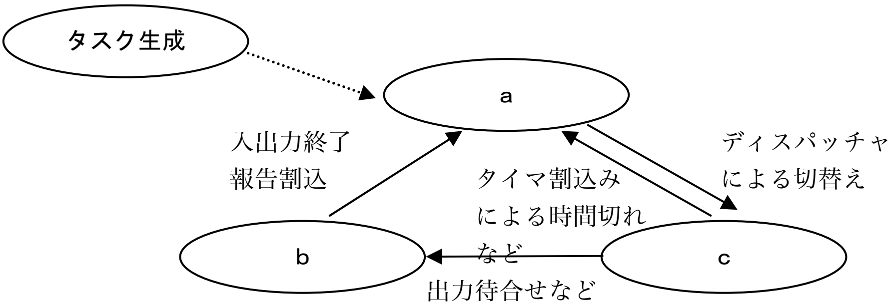


図　タスクの状態遷移

	a	b	c
ア	実行可能状態	実行状態	待ち状態
イ	実行可能状態	待ち状態	実行状態
ウ	実行状態	実行可能状態	待ち状態
エ	実行状態	待ち状態	実行可能状態
オ	待ち状態	実行状態	実行状態

Copyright 2005 BohYoh Shibata

第２種【平成 10 年度春】午前･問 33

プロセスの起動から終了までの状態遷移として、あり得ないものはどれか。

ア　起動→実行可能状態→実行状態→終了
イ　起動→実行可能状態→待機状態→実行状態→終了
ウ　起動→実行可能状態→実行状態→実行可能状態→実行状態→終了
エ　起動→実行可能状態→実行状態→待機状態→実行可能状態→実行状態→終了

第１種【平成 9 年度】午前･問 26

プロセスの状態は、実行、実行可能、待機の三つを基本と考えることができる。実行状態にあるプロセスがプリエンプションによって他のプロセスに実行を中断され､再び実行状態に戻るまでの状態遷移を表したものはどれか。

ア　実行→実行可能→実行
イ　実行→実行可能→待機　→実行
ウ　実行→待機　→実行
エ　実行→待機　→実行可能→実行

覚えておきたい Windows ショートカット

⊞＋ E　マイコンピュータ（Explorer）を起動する。
⊞＋ F　検索ウィンドウを開く。
⊞＋ L　コンピュータをロックする。
⊞＋ M　すべてのウィンドウを最小化する。

Copyright 2005 BohYoh Shibata

■ プロセス切替えの原理 ■

複数のプロセスを見かけ上同時に動かすためには、時間の流れを区切り、それぞれのプロセスに CPU を割り当てなければなりません。

プロセスの切替え方式としては、大きく二つの方法があります。

１　イベントドリブン（事象駆動 *event driven*）

事象すなわちイベント（システムの状態変化）が発生したのを契機にして、CPU を切り換えてタスクのスケジューリングを実行する方式です。マルチプログラミングを実現するための基本となる技術です。

イベント（*event*）が発生するタイミングの例

(a) 入出力の完了
(b) 入出力要求の発生
(c) ジョブの到着
(d) ジョブの完了

※　イベントが生じたときに、実行中のプロセスを一時停止し、そのプロセスの再続行が可能となるように CPU の状態を保持する**割込み**（*interrupt*）機構が必要となります。

２　タイムスライシング（時分割 *time-slicing*）

システムの状態変化とは無関係に、設定した短い時間＝**タイムクオンタム**（*time-quantum*）の周期でプロセスを切り換える方式。TSS の実現には必要不可欠な技術です。

※　クオンタムが終了したら割込を生じさせる**インタバルタイマ**が必要となります。

＊

マルチプログラミングの実現は、イベントドリブン方式でのみ可能であり、TSS の実現にはタイムスライシングの機能が必要です。多くのシステムは、マルチプログラミングとタイムシェアリングの両方のサービスを備えており、これら２方式を組み合わせて利用します。

なお、タイムスライシングの導入は、マルチプログラミングシステムの性能を向上させ、イベントドリブンの導入は、タイムシェアリングシステムのリソース利用効率を向上させることが分かっています。

Copyright 2005 BohYoh Shibata

第２種【平成９年度秋】午前・問 34

CPU の処理時間を微小時間に分割し、それを実行可能な状態にあるタスクに割り当てる形態はどれか。

ア　オーバレイ　　イ　スワッピング
ウ　タイムスライシング　　エ　リアルタイム処理

第１種【平成８年度】午前・問 28

イベントドリブンスケジューリングに関する記述のうち、正しいものはどれか。

ア　環境の変化をトリガとして、タスクのスケジューリングを実行する。
イ　タスクの優先順位をダイナミックに変更しながらスケジューリングをする方式である。
ウ　タスクを一定周期で実行するためのスケジューリング方式である。
エ　優先度の低いタスクのスケジューリングに適用する。
オ　ラウンドロビン方式と組み合わせて使うことができないスケジューリング方式である。

第１種【平成 10 年度】午前・問 24

イベントドリブン方式のプロセス切替えに関する記述のうち、正しいものはどれか。

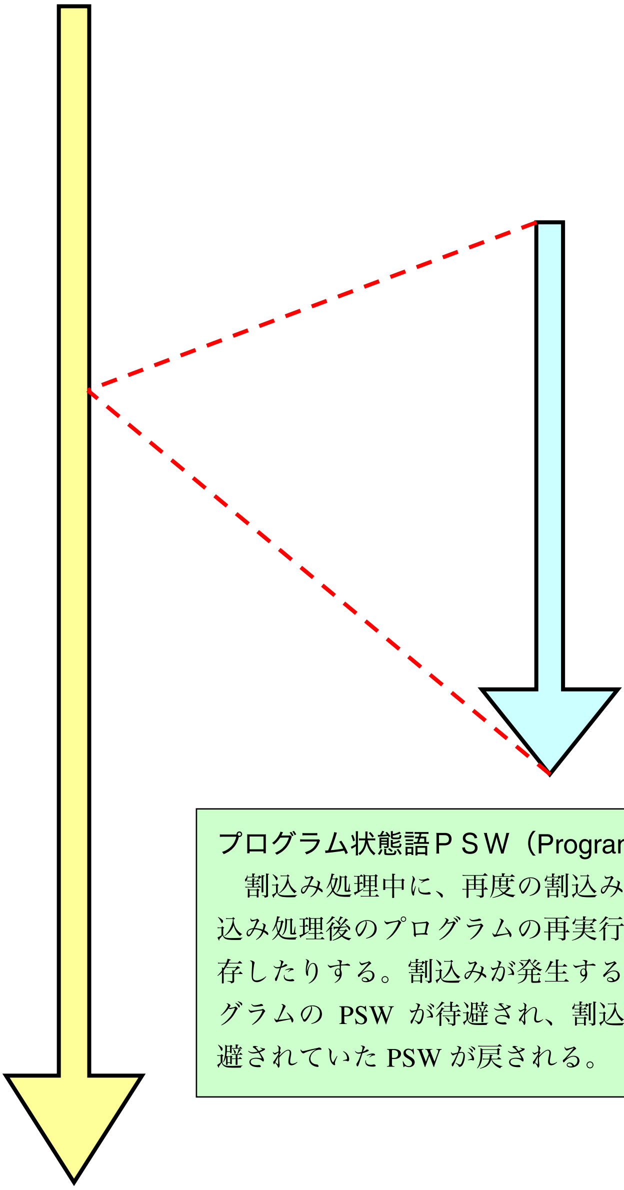
ア　イベントドリブン方式とタイムスライシング方式は排他的なものであり、一つのシステムで共存することはない。
イ　プロセス切替えのきっかけは、入出力終了やプログラム異常などであり、ユーザプログラムが要求するものではない。
ウ　プロセス切替えの処理はソフトウェアによって行われ、ハードウェアが介在することはない。
エ　割り込まれたプロセスが、割込み処理終了後に、引き続いて実行される保証はない。

Copyright 2005 BohYoh Shibata

割込み（interrupt）

実行中のプロセスを一時停止し、そのプロセスの再続行が可能なように CPU の状態などを保存する。割込みは、一種のサブルーチンコール（関数呼び出し）のようなものであり、**割込み条件**が成立すると、割込みの種類に応じてあらかじめ指定されたプログラムが実行される。実行後は（何もなかったかのように）元のプログラムに戻る。

ただし、割込みが発生しても、その原因によらず無条件に受け付けられるとは限りません。



プログラム状態語ＰＳＷ（Program Status Word）

割込み処理中に、再度の割込みを防止したり、割込み処理後のプログラムの再実行に備えた情報を保存したりする。割込みが発生すると、実行中のプログラムの PSW が待避され、割込み処理終了後に待避されていた PSW が戻される。

Copyright 2005 BohYoh Shibata

割込みは、以下のような条件が成立したときに発生します。

割込み条件

入出力動作が完了した

インタバルタイマの時間が０になった

ハードウェア異常を検出した

演算異常（オーバーフロー）を検出した

（ａ）**スーパーバイザコール割込み**（*supervisor call interrupt*）

実行モード

特権モード（*master mode*／*supervisor mode*）

非特権モード（*slave mode*／*problem mode*）

非特権モードで実行しているプログラムは、オペレーティングシステムの核である**ニュークリアス**（*nucleus*）のサービスの要求を SVC 命令によって行います。

（ｂ）**入出力割込み**（*input/output interrupt*）

入出力操作が完了したときや、誤った入出力を指定した場合などに、入出力チャネルにより発生させられます。

（ｃ）**プログラムチェック割込み**（*program check interrupt*）

オーバフロー、０による除算、記憶保護違反などによって発生します。

（ｄ）**マシンチェック割込み**（*machine check interrupt*）

以下に示すような、ハードウェアの障害によって生じます。

・誤動作

・故障

・電源のトラブル　など

（ｅ）**外部割込み**（*external interrupt*）

インタバルタイマの終了、他のコンピュータシステムからの信号などから生じます。

（ｆ）**リスタート割込み**（*restart interrupt*）

リセットボタンを押したときなどに発生します。

Copyright 2005 BohYoh Shibata

多重割込み

割込み処理の実行中に割込みが発生することを**多重割込み**といいます。ただし、多重割込みには優先順位が付けられており、高い優先順位の割込み処理の実行中は、優先順位の低い割込みはマスクされます。

割込みの優先順位は、通常は、以下のようになっています。

優先順位	割込みの原因
1	機械チェック
2	外部割込み
3	入出力割込み（入出力処理の終了）
4	ＳＶＣ／プログラム割込み

第２種【平成７年度秋】午前･問 25

割込みに関して、正しい記述はどれか。

ア　タイマ割込みは、内部割込みの一種である。
イ　入出力割込みは、制御装置と入出力装置を並行動作させるために使われる。
ウ　割込みが発生すると、次の割込みまで、制御装置は停止する。
エ　割込みは、実行順序を強制的に変更する分岐命令である。
オ　割込みは、ハードウェアによって発生するので、ソフトウェアで発生させることはできない。

第２種【平成 10 年度秋】午前･問 34

割込みに関する記述として、正しいものはどれか。

ア　入出力割込みによって、処理装置と入出力装置の並行動作が可能となる。
イ　割込みが発生すると、次の割込みまで処理装置は停止する。
ウ　割込みは、その原因によらず無条件に受け付けられる。
エ　割込みは、ハードウェアによって発生するものであり、ソフトウェアでは発生させることができない。

第２種【平成９年度春】午前･問 29

プログラム割込みの原因となる得るものはどれか。

ア　入出力動作が終了した。
イ　ハードウェアが故障した。
ウ　プログラムで演算結果があふれた（オーバーフローした）。
エ　プログラムの実行時間が設定時間を超過した。

Copyright 2005 BohYoh Shibata

第2種【平成12年度秋】午前･問27

プログラム割込みの原因となり得るものはどれか。

ア　入出力動作が終了した。

イ　ハードウェアが故障した。

ウ　プログラムで演算結果があふれた（オーバフローした）。

エ　プログラムの実行時間が設定時間を超過した。

第2種【平成11年度春】午前･問20

内部割込みの原因となるものはどれか。

ア　コンピュータの電源装置における異常の発生

イ　処理装置内で時間計測を行うカウンタの所定値越え

ウ　入出力装置の動作終了や障害発生

エ　浮動小数点演算でのけたあふれ（オーバフロー）の発生

第2種【平成7年度春】午前･問23

入出力に関する次の記述中の［　　］に入れるべき適切な字句の組合せはどれか。

a　が処理装置に代わって入出力処理を実行することによって、処理装置の動作と入出力動作のオーバーラップが可能となり、全体の効率が向上する。一方、入出力の終了は　b　機能によって処理装置に伝えられ、処理装置の動作と入出力動作の同期がとられる。

	a	b
ア	制御装置	機械チェック割込み
イ	制御装置	入出力割込み
ウ	制御装置	プログラム割込み
エ	チャネル	機械チェック割込み
オ	チャネル	入出力割込み

第1種【平成8年度】午前･問21

割込みに関する記述のうち、正しいものはどれか。

ア　割込み処理が、更に割り込まれることはない。

イ　割込み処理は、先に発生した割込みから順番に行われる。

ウ　割込みは禁止することができる。

エ　割込みは、入出力の終了を知らせるためだけに用いられる。

エ　割込みを明示的に発生させるための命令はない。

第1種【平成10年度】午前･問98

リアルタイムOSが行う多重割込みの処理方法として、適切なものはどれか。

ア　現在処理をしている割込み処理よりも、優先順位の高い割込みをマスクをする。

イ　現在処理をしている割込み処理よりも、優先順位の低い割込みをマスクをする。

ウ　現在処理をしている割込み処理を中断して、後から発生した割込み処理を優先する。

エ　現在処理をしている割込み処理を優先して、後から発生した割込み処理を待たせる。

Copyright 2005 BohYoh Shibata

第 1 種【平成 11 年度】午前･問 99

リアルタイム OS におけるタスクの優先順位に関する記述として、正しいものはどれか。

ア　I/O の完了やタイマの完了、タスク間の連絡など異なったイベントで待ち状態になったタスクが複数ある場合、タスクは優先順位に従って実行可能状態になる。

イ　システムのデバイスドライバから I/O の完了を通知されたタスクは、優先順位に関係なく一定時間実行状態になる。

ウ　セマフォを使用して共有資源を占有していても、優先順位の高いタスクが同じ資源を要求すると、優先順位の低いタスクは資源の占有を解かれてしまう。

エ　優先順位の低いタスクは、長い時間実行可能状態にあっても、優先順位のより高いタスクが実行状態にある間は実行状態になれない。

第 1 種【平成 12 年度】午前･問 21

表に示す 3 種類の割込み種別をもち，多重割込みを許すリアルタイムシステムがある。割込み C が発生した 20 ミリ秒後に割込み A が発生し，割込み A 発生後 80 ミリ秒後に割込み B が発生した。この場合，割込み C の処理が終了するまでの時間は何ミリ秒か。ここで，割込みの優先度は A ＞ B ＞C とし，OS などのオーバヘッドは無視するものとする。

割込み種別	周期（ミリ秒）	割込み処理時間（ミリ秒）
A	50	10
B	非同期	30
C	非同期	85

ア　105　　イ　135　　ウ　145　　エ　155

Copyright 2005 BohYoh Shibata

■ ＣＰＵスケジューリング ■

ＣＰＵはプロセスの間で切り替えられ、その方式がスケジューリングであり、多種多様な方法が存在する。

到着順　FIFO/FCFS（*First-In First-Out / First-come First-served*）
処理時間順　SPT（*shortest-processing time-first*）
優先度順　（*priority scheduling*）
ラウンドロビン　RR（*round robin*）
フィードバック持ち行列　FB（*feed back queue*）
：

性能評価基準

(ａ)応答時間（*turnaround time／response time*）

(ｂ)スループット（*throughtput*）

・予測性

・公平性

・グレースフルな終了

第２種【平成12年度秋】午前・問30

システムが単位時間内にジョブを処理する能力の評価尺度はどれか。

ア　MIPS値　　イ　応答時間　　ウ　スループット　　エ　ターンアラウンドタイム

Copyright 2005 BohYoh Shibata

（1）到着順　FIFO（*Fisrt-In Fisrt-Out*）／FCFS（*First-come First-served*）

ジョブを到着順に実行する。ジョブはいったん開始されると完了まで実行されます。すなわち、ノンプリエンプティブです。

ジョブの到着

○ … ○ ○ ○　○ CPU

公平かつ単純であるが、あまり優れた方法であるとは言えない。
応答時間が長くなるため TSS には適さない。
本来優先すべきタスクであっても、先に実行されるとは限らない。

※MS-Windows3.0 で採用されていました。

Copyright 2005 BohYoh Shibata

（２）処理時間順　SPT（*shortest-processing time-first*）／SJF（*shortest job first*）

処理時間の短いジョブから実行します。平均応答時間を最小にし、利用効率を向上させやすいという特徴があります。ノンプリエンプティブな方式です。

ジョブの到着

○ …　○ ○ ○ ○　○ CPU

予め処理時間を知ることは困難である。
プリエンプションと組み合わせると効果的。
実行を一時中断し、他のジョブを実行する。

Copyright 2005 BohYoh Shibata

（３）優先度順　（*priority scheduling*）

各プロセスに優先度を与え、優先度の順に実行する方式です。

ジョブの到着

○ … ○ ○ ○　○ ［ＣＰＵ］

より優先度の高いジョブが到着するか、
待ち状態のジョブがより高い優先度を得ると、
現在のジョブは中断される。

優先順位：

1.リアルタイムプロセス
2.オペレーティングシステム
3.対話型ジョブ
4.バッチジョブ（バックグラウンドジョブ）

静的優先度方式（最初に与えられた優先度は変化しない）

動的優先度方式（優先度が状況に応じて動的に変化する）

優先度の低いプロセスが待たされる**スタベーション**（*starvation*）が発生しやすい 対策：**エージング**（*aging*）

Copyright 2005 BohYoh Shibata

（4）ラウンドロビン　RR（*round robin*）

各プロセスに、タイムクォンタムを割り当てて、そのプロセスがその時間内に終了しないと、実行可能行列の末尾に移されます。対話型ジョブの応答時間を短くします。

○ … ○ ○ ○　○ ［ＣＰＵ］

quantum を消費すると、現在のジョブは中断され、待ち行列の末尾に戻される。

quantum を無限大にすると FCFS と同じ‥‥。

Copyright 2005 BohYoh Shibata

（５）フィードバック待ち行列　FB（*feed back queue*）

多段のＲＲです。新しく到着したプロセスは優先度の最も高い待ち行列に追加します。
処理が終了せずに中断されたジョブは、一つ下の優先度の待ち行列に追加されます。
　→処理時間の長いジョブは、最も低い優先度の待ち行列に落ちる。

○ … ○ ○ ○　○　ＣＰＵ

レベル１
○ … ○ ○ ○　○

レベル２
○ … ○ ○ ○　○

レベルｎ
○ … ○ ○ ○　○

選ばれたジョブは、最後まで実行されます。

スタベーションが発生しやすいので、エージングを併用します。

（６）入出力ジョブ優先方式

入出力の割合の大きいジョブを優先します。

入出力の割合の大きいジョブを優先します。
- ＣＰＵ処理のブロッキングをさけます。
- 入出力の仕事は、ユーザにとって次のジョブの準備につながる傾向にあります。

Copyright 2005 BohYoh Shibata

（Ｉ）応答時間の評価

３つのジョブが、以下のように与えられる。応答時間を求めよ。

	到着時刻	処理時間
ジョブ１	0	１０
ジョブ２	2	２０
ジョブ３	6	5

(1)　到着順

ジョブ１																		
ジョブ２																		
ジョブ３																		

完了時刻	応答時間

(2)　ラウンドロビン（quatum = 1 sec）

ジョブ１																		
ジョブ２																		
ジョブ３																		

完了時刻	応答時間

(3)　処理時間順

ジョブ１																		
ジョブ２																		
ジョブ３																		

完了時刻	応答時間

Copyright 2005 BohYoh Shibata

（II）処理時間・利用効率の評価

以下のスケジューリングを考え、処理時間と利用効率を計算します。

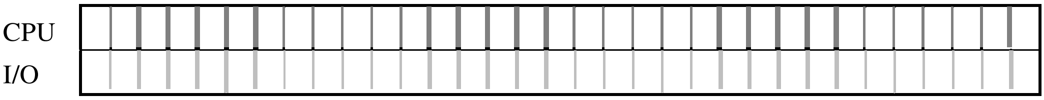
・処理開始時直前に、ジョブ１→ジョブ２→ジョブ３の順にジョブが投入されています。

・各処理において、ＣＰＵ処理が完了した後に、入出力処理が行われます。

・三つのジョブを一連の作業と考えると、この作業は２回行われます。

	CPU 処理	入出力処理
ジョブ１	35	10
ジョブ２	20	20
ジョブ３	5	25

(1)　到着順



CPU の利用効率　＝

I/O の利用効率　＝

平均待ち時間　　＝

(2)　処理時間順

CPU の利用効率　＝

I/O の利用効率　＝

平均待ち時間　　＝

Copyright 2005 BohYoh Shibata

第1種【平成11年度】午前･問23

ノンプリエンプティブなマルチタスク OS の欠点はどれか。

ア　アプリケーションプログラムがループすると OS に制御が戻らない。

イ　仮想記憶のページ置換えの機能が不十分となる。

ウ　カーネルモードとユーザモードの区分がなく、OS が破壊される危険性がある。

エ　割込み処理が遅れ、入出力処理のリアルタイム性が劣る。

ソフトウェア【平成13年度】午前･問26

リアルタイム OS のプリエンプションの記述として、適切なものはどれか。

ア　一定時間ごとに実行中のタスクを中断して、別の実行可能なタスクを実行する。

イ　実行中のタスクが終了または待ち状態になった場合、別の実行可能なタスクを実行する。

ウ　実行中のタスクよりも優先順位の高いタスクが実行可能状態になった場合、実行中のタスクを中断してその優先順位の高いタスクを実行する。

エ　優先順位の高いタスクが優先順位の低いタスクの終了を待っている場合、優先順位の低いタスクの優先順位を一時的に高くして実行する。

第1種【平成11年度】午前･問98

タスクが実行状態（RUN）、実行可能状態（READY）、待ち状態（WAIT）の三つの状態で管理されるリアルタイムＯＳにおいて、タスクＡ，Ｂ，Ｃがプリエンプティブなスケジューリングによって、図に示すとおりに状態遷移した。各タスクの優先順位の関係のうち、正しいものはどれか。ここで、優先順位の関係は“高い＞低い”で表す。

<table>
<tr><td>タスクＡ</td><td>RUN</td><td colspan="2">WAIT</td><td>READY</td><td>RUN</td><td>READY</td></tr>
<tr><td>タスクＢ</td><td>WAIT</td><td>RUN</td><td>WAIT</td><td>RUN</td><td colspan="2">WAIT</td></tr>
<tr><td>タスクＣ</td><td>WAIT</td><td>READY</td><td>RUN</td><td colspan="2">WAIT</td><td>RUN</td></tr>
</table>

ア　Ａ＞Ｂ＞Ｃ　　イ　Ｂ＞Ａ＞Ｃ　　ウ　Ｂ＞Ｃ＞Ａ　　エ　Ｃ＞Ｂ＞Ａ

第1種【平成10年度】午前･問23

プロセススケジューリングに関する記述のうち､ラウンドロビン方式の説明として正しいものはどれか。

ア　各プロセスに優先順位が付けられる。後に到着したプロセスの優先順位が処理中のプロセスよりも高ければ、処理中のものを中断し、到着プロセスを処理する。

イ　各プロセスの処理時間に比例して、タイムクウォンタム（CPU 割当て時間）を変更する。

ウ　各プロセスを待ち行列に置かれた順にタイムクウォンタムずつ処理し、終了しないときは同じ優先順位の待ち行列の最後尾につなぐ。

エ　入出力の終了やコマンド入力などの割込みを引き金にして、次に実行するプロセスを決める。

Copyright 2005 BohYoh Shibata

第１種【平成７年度】午前・問26

オペレーティングシステムにおけるタスクのスケジューリングに関して、正しいものを二つ選べ。

ア　FIFO（First-In First-Out）方式では、要求された順番にCPU時間を割り当てるので、処理時間の短いタスクほど順番が有利となる。
イ　RR（Round Robin）方式は、要求された順番にCPU時間を割り当て、割り当てられた時間を使い切ったあとは、待ち行列の最後に回す方式である。
ウ　SJF（Shortest Job First）方式は、JCLなどで指定した処理時間の短いタスクを先に処理する方式で、処理時間の長いタスクほど後回しになる。
エ　多重待ち行列方式は、最初に低い優先順位と長いCPU時間を割り当て、その後は優先度を高くし、CPU時間を徐々に短くする方式である。
オ　優先度順（Priority）方式は、各タスクに優先順位を与えて、順位の高い順にCPU時間を割り当てるので、優先度の低いタスクでもCPU使用率が高いと優先度が上がり、不利にならない方式である。

ソフトウェア【平成13年度】午前・問25

OSにおけるタスクのスケジューリングに関する記述として、適切なものはどれか。

ア　多重待ち行列方式は、割り当て要求のあったタスクに対して最初に低い優先順位と長いCPU時間を割り当て、その後は優先度を高くし、CPU時間を徐々に短くする方式である。
イ　到着順方式では、タスクが生成された順に高い優先順位を付けてCPU時間を割り当てる。これは先に開始されたタスクを優先させて、早く終了させることを目的としているからである。
ウ　優先順位方式では、CPUの利用状況の低いタスクの優先順位を順次高くし、逆にCPUを多く利用したタスクの優先順位を低くするので、システム全体の処理効率を高めるのに適している。
エ　ラウンドロビン方式は、要求された順番にCPU時間を割り当て、割り当てられた時間を使い切った後は、待ち行列の末尾に回す方式である。

第１種【平成12年度】午前・問25

プロセスのスケジューリング方式におけるフィードバック待ち行列方式に関する記述として、適切なものはどれか。

ア　一定時間内に処理が終了しない場合は、順次優先度を落としていく方式である。
イ　実行待ちリストで待っているプロセスの中で、推定実行時間が最も短いものを次に選ぶ方式である。
ウ　実行待ちリストに到着した順に従って、プロセスを実行する方式である。
エ　プロセスを到着順に実行し、一定時間内に終了しない場合は、実行待ちリストの最後尾につなぐ方式である。

Copyright 2005 BohYoh Shibata

第２種【平成８年度春】午前・問 55

Ａ及びＢの二つのプログラムがあり、それぞれ単体で実行したときの処理装置（CPU）、入出力装置（I/O）の占有時間は、図の通りである。

プログラムＡ、 Ｂを 1 台の CPU のもとで起動したとき、プログラムＢが最短で終了するのは起動の何ミリ秒後となるか。ここで、プログラム等の実行条件は次の通りとする。

① プログラムの実行優先度はＡのほうがＢより高い。
② プログラムＡ、Ｂは同一の入出力装置を利用する。
③ CPU 処理を実行中のプログラムは、入出力装置を行うまでは実行を中断されない。
④ 入出力装置も入出力処理が終了するまで実行を中断されない。
⑤ CPU 処理の切替え（タスクスイッチ）に必要な時間は無視できる。

プログラムＡ　　　　　　　　　　　ミリ秒

CPU	I/O	CPU	I/O	CPU
20	30	20	40	10

プログラムＢ　　　　　　　　　　　ミリ秒

CPU	I/O	CPU	I/O	CPU
10	30	20	20	20

ア　100　　イ　120　　ウ　140　　エ　160　　オ　180

Copyright 2005 BohYoh Shibata

第２種【平成12年度春】午前・問35

三つのタスクの優先度及び各タスクを単体で実行した場合の処理装置（CPU）、入出力装置（I/O）の占有時間は、表のとおりである。三つのタスクが同時に実行可能状態になってから、すべてが終了するまでのCPUのアイドル時間は何ミリ秒か。ここで、CPUは1個とし、各タスクのI/O処理は並行して処理可能であり、OSのオーバヘッドは無視できるものとする。

<table>
<tr><td>タスク</td><td>優先度</td><td>単独動作時の所要時間</td></tr>
<tr><td>A</td><td>高</td><td rowspan="3">各タスクともに、CPU 5ミリ秒 →I/O 8ミリ秒→CPU 2ミリ秒</td></tr>
<tr><td>B</td><td>中</td></tr>
<tr><td>C</td><td>低</td></tr>
</table>

ア　3　　イ　4　　ウ　5　　エ　6

Copyright 2005 BohYoh Shibata

第２種【平成12年度秋】午前・問32

　三つのタスクの優先度、各タスクを単体で実行した場合の処理装置（CPU）と入出力装置（I/O）の占有時間は、表のとおりである。優先順位方式のタスクスケジューリングを行うOSのもとで、三つのタスクが同時に実行可能状態になってから、タスクCが終了するまでの間に、タスクCが実行可能状態にある時間は延べ何ミリ秒か。ここで、各タスクの入出力は並行して処理が可能であり、OSのオーバヘッドは無視できるものとする。

タスク	優先度	単独実行の占有時間（単位：ミリ秒）				
		CPU • I/O • CPU • I/O • CPU				
A	高	4	4	3	5	3
B	中	2	6	3	6	2
C	低	2	5	3	4	1

ア　2　　イ　5　　ウ　8　　エ　11

第１種【平成８年度】午前・問 29

プロセスＡ，Ｂはそれぞれ次のような時間（ミリ秒）で CPU と I/O（入出力装置）を用いた処理をする。

プロセスＡ：CPU(10) → I/O(20) → CPU(10) → I/O(30) → CPU(10)

プロセスＢ：CPU(5) → I/O(30) → CPU(5) → I/O(20) → CPU(5)

（→はプロセスＡ，Ｂの進行過程を示し、( )内の数字は処理時間を示す。）

この二つのプロセスを同時に実行させたとき、プロセスの開始から二つのプロセスの完了まで間の CPU 利用率（%）に最も近いものはどれか。ここで、同期実行の条件は次のとおりとする。

- 最初はプロセスＡに CPU 時間が割り当てられる。
- CPU 時間は 5 ミリ秒ごとのラウンドロビン方式でスケジューリングされる。
- 二つのプロセスがアクセスする入出力装置は異なった装置であり、同時処理が可能である。
- 割込み処理に要する時間は無視してよい。
- この二つのプロセスだけが実行され、ほかにプロセスはない。

ア　40　　イ　45　　ウ　50　　エ　55　　オ 60

Copyright 2005 BohYoh Shibata

第１種【平成 11 年度】午前・問 24

三つのジョブＡ～Ｃを、次のスケジューリング方式に基づいて同時に開始する。このとき、ジョブＢが終了するまでの経過時間は、ほぼ何分か。

［ジョブ］

単独で実行したときの処理時間は、ジョブＡは 5 分、ジョブＢは 10 分、ジョブＣは 15 分である。処理時間内の内訳はすべて CPU 時間である。ジョブの切替え時間は無視できる。

［スケジューリング方式］

（１）待ち行列に並んだ順に実行する。

（２）一定時間（これをタイムウォンタムと呼ぶ）の間に処理が終了しなければ、処理を中断させて、待ち行列の最後尾へ回す。

（３）タイムクウォンタムは、ジョブの処理時間に比べて十分に小さい値とする。

ア　15　　イ　20　　ウ　25　　エ　30

ソフトウェア【平成 13 年度】午前・問 36

（同じ問題）

Copyright 2005 BohYoh Shibata

第１種【平成 10 年度】午前・問 78

　プロセッサと入出力チャネルで構成されるシステムで、表に示すジョブA，B，Cを実行した。すべてのジョブが実行されるまでのプロセッサ及び入出力チャネル使用率として、正しいものはどれか。ここで、ジョブA，B，Cはシステムの動作開始時点ではいずれも実行可能状態にあり、この順序で実行される。入出力チャネルは、入出力要求を発生順に処理する。ジョブA，B，Cとも、プロセッサの処理を終了した後、入出力チャネルの処理を実行する。

単位　秒

ジョブ	プロセッサの処理時間	入出力チャネルの処理時間
A	35	10
B	20	20
C	5	25

	プロセッサ使用率	入出力チャネル使用率
ア	0.52	0.48
イ	0.60	0.55
ウ	0.75	0.69
エ	1.00	0.92

Copyright 2005 BohYoh Shibata

■ プロセスとスレッド

・プロセス

プロセスは、本来 UNIX のシステムに関する用語で、実行中のプログラムのアドレス空間など、実行環境を含むものです。アドレス空間を分割して実行できるため、並行処理であるパイプ処理やバックグラウンド処理にも適しています。しかし、各プロセスが別個のアドレス空間を利用するために多くの資源が必要である、生成や切替えに時間がかかるなどの欠点があります。そのため**重量プロセス**（**ヘビーウェイトプロセス**）とも呼ばれます。

・スレッド

上記のようなプロセスの問題を解決するのが、**軽量プロセス**（**ライトウェイトプロセス**）とも呼ばれる**スレッド**（*thread*）です。スレッドは、プロセスを細分化した基本的な処理単位です。高速で生成できる、スワップの負荷が小さいなどの特徴がある。CPU 資源のみが割り当てられて、スタック、プログラムカウンタ、レジスタセットを保持しますが、その他の資源は親のプロセスから継承します。同一のプロセスから発生するスレッドは、同じアドレス空間を継承し共有しなければなりません。分散処理や、マルチ CPU での並行処理などで利用されます。

第１種【平成８年度】午前･問 26

プログラムの実行に必要なシステム資源の割当ての単位であるプロセスとは異なり､軽量プロセスとも呼ばれ、CPU 以外の資源は割り当てられず、親のプロセスから必要な資源を継承するものはどれか。

ア　カーネル　　イ　コンテキスト　　ウ　ジョブステップ
エ　スレッド　　オ　タスク

第１種【平成 10 年度】午前･問 25

並行処理の単位として、プロセスのほかに、プロセス内に複数存在するスレッドを用いることがある。一つのプロセス内のスレッドが共有するものはどれか。

ア　アドレス空間　　イ　スタック
ウ　プログラムカウンタの値　　エ　レジスタセットの値

Copyright 2005 BohYoh Shibata

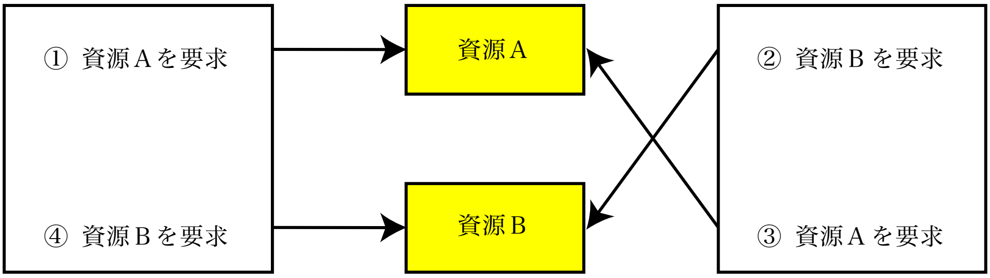
セマフォ

セマフォとは、手旗信号を意味する言葉です。コンピュータの世界では、E.W.Dijkstar が提案した、カウンタと待ち行列からなるデータ構造を指し、プロセスの排他制御のために用いられます。

資源の状態を待ち（P＝資源獲得)、続行（V＝資源解放）の二つの操作で表すことで、共有データの排他的アクセス、同一資源の割当てなどの作業を行い、プロセス間で同期を取って、資源を共有する複数のプロセスの処理を円滑に行います。

デッドロック

デッドロック（*dead lock*）とは、複数のプロセスが互いに次にアクセスする資源にロックをかけてしまい、永久に待ちの状態に陥ることです。



第２種【平成９年度春】午前・問 33

プロセスの相互排除（排他制御）に用いられるものはどれか。

ア　コンテンション　　　　イ　セマフォ
ウ　チェックポイント　　　エ　ハッシュ

第２種【平成 12 年度秋】午前・問 29

プロセスの相互排除（排他制御）に用いられるものはどれか。

ア　コンテンション　　　　イ　セマフォ
ウ　チェックポイント　　　エ　ハッシュ

第１種【平成９年度】午前・問 27

セマフォを用いる目的として、正しいものはどれか。

ア　共有資源を管理する。　　　イ　スタックを容易に実現する。
ウ　スラッシングを減らす。　　エ　セグメンテーションを実現する。

Copyright 2005 BohYoh Shibata

第1種【平成7年度】午前･問28

オペレーティングシステムの機能である同期制御に関する記述として、正しいものはどれか。

ア　Post / Wait マクロ命令は、リソースの占有使用要求及び解放に用いる。
イ　セマフォは、タスクの同期を取るために用いる。
ウ　ダブルバッファリング方式は、タスクの同期制御を実現する一つの手段である。
エ　データ指向モデルは、並行処理の動作を簡潔に表現できるモデルである。

第1種【平成11年度】午前･問21

次のケースのうち、デッドロックが起こるものはどれか。

ア　タスクＸが資源Ａと資源Ｂを確保した後に、タスクＹが資源Ａと資源Ｂを共有した。
イ　タスクＸが資源Ａを確保し、タスクＹが資源Ｂを確保した後に、タスクＹが資源Ａを要求し、タスクＸが資源Ｂを要求した。
ウ　タスクＸが資源Ａを確保し、タスクＹが資源Ｂを確保した後に、タスクＹが資源Ｃを要求し、タスクＸが資源Ｂを要求した。
エ　タスクＸが資源Ａを確保し、タスクＹが資源Ｂを確保した後に、タスクＺが資源Ａと資源Ｂを要求した。

第1種【平成9年度】午前･問28

資源Ｘ，Ｙ，Ｚを占有して処理を行うプロセスＡ～Ｄがある。各プロセスは処理の進行に伴い、表中の数値の順に資源を占有し、実行終了後に三つの資源を一括して解放する。プロセスＡとデッドロックの関係を起こす可能性のあるプロセスはどれか。

プロセス名	資源の占有順序		
	資源Ｘ	資源Ｙ	資源Ｚ
プロセスＡ	1	2	3
プロセスＢ	3	2	1
プロセスＣ	2	3	1
プロセスＤ	1	2	3

ア　プロセスＢだけ　　イ　プロセスＣだけ
ウ　プロセスＢ，Ｃ　　エ　プロセスＢ，Ｃ，Ｄ

Copyright 2005 BohYoh Shibata

ファイル管理

データの長期保存は、OS の重要な役割の１つであり、**ファイル**（*file*）と呼ばれる形式で保存を行います。

ディレクトリ

膨大な数のファイルを１元的に管理するのは困難であるため、UNIX や Windows95/NT などの OS では、階層的な構造を持つ**ディレクトリ**（*directory*）の概念が導入されています。ディレクトリは、複数のファイルをまとめてグループ化するためのものであると考えましょう。ディレクトリは、階層的な構造をもつことができます。

※　最近の Windows では、ディレクトリでなく“フォルダ”と呼ばれます。

■ ルートディレクトリ（root directory）

最も上位に位置するディレクトリであり、１個のみが存在します。

UNIX では/と表し、日本語版 Windows では¥と表します。

■ サブディレクトリ（sub directory）

階層的なディレクトリ構造において、下位に位置するディレクトリを**サブディレクトリ**あるいは**子ディレクトリ**と呼びます。

■ 親ディレクトリ（super directory）

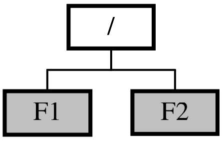
上位に位置するディレクトリ。ルートディレクトリ以外のディレクトリは、必ず１個の親ディレクトリを持ちます。

※　ルートディレクトリを含め、各ディレクトリは、任意の個数（0 個以上）のファイルおよび任意の個数のサブディレクトリを持つことができます。

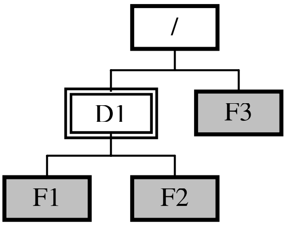
Copyright 2005 BohYoh Shibata

■ ディレクトリとファイルの例

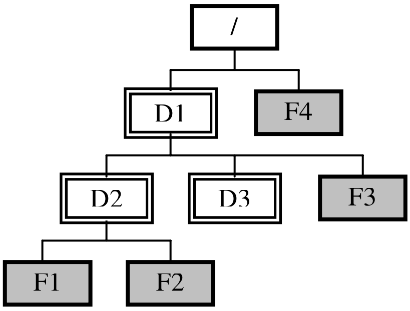
ファイルもサブディレクトリも存在しない。



ルートディレクトリの下にファイル F1 および F2 が存在。サブディレクトリは存在しない。



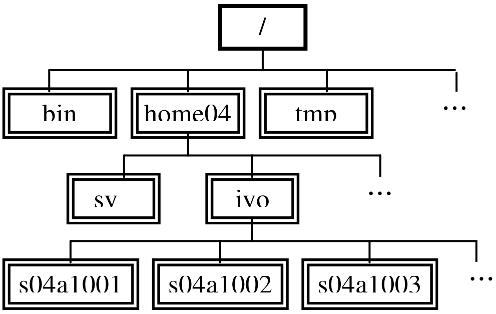
ルートディレクトリの下にディレクトリ D1 およびファイル F3 が存在。サブディレクトリ D1 の下には、ファイル F1 および F2 が存在。
※ディレクトリ D1 の親ディレクトリは、ルートディレクトリ



ディレクトリ D3 の下は空。
※ディレクトリ D2 および D3 の親ディレクトリは、いずれも D1。

■ 本学でのディレクトリ構成

本学センターでは、ルートディレクトリの下に、多くのサブディレクトリが存在する。2004 年 4 月入学の学生用のサブディレクトリが、**home04** であり、その下に、各学科用のサブディレクトリが用意されています。情報工学科用のディレクトリは **jyo** であり、その下に各個人用のディレクトリ **s04a1001**, **s04a1002**, **s04a1003**…が存在します。



Copyright 2005 BohYoh Shibata

パス

任意のファイルやディレクトリを表す方式として、以下に示す二つがあります。

■ 絶対パス

ルートディレクトリからの全ての経路で表します。

■ 相対パス

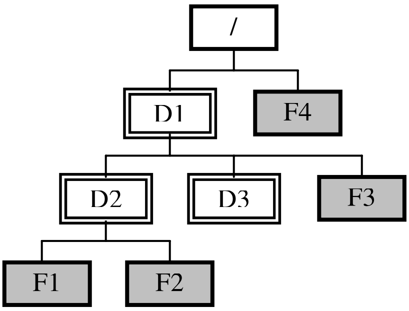
現在作業を行っている**カレントディレクトリ**（*current directory*）／**ワーキングディレクトリ**（*working directory*）からの経路で表します。

パスの表現では、以下の記号を利用します。

`..`	親ディレクトリを表します。
`/`	パス表現の先頭にある場合はルートディレクトリを、 中間にある場合は、ディレクトリ名又はファイル名の区切りを表します。 ※日本語版 Windows では¥を用います。

※ 絶対パスは、ルートディレクトリからの経路で表すため、カレントディレクトリとは無関係に決定します。

※ 相対パスは、カレントディレクトリからの経路で表すため、カレントディレクトリに依存します。



左図において、各ファイルの絶対パス、相対パスを考えてみましょう。

絶対パス

ファイル F1	`/D1/D2/F1`
ファイル F3	`/D1/F3`
ファイル F4	`/F4`
ディレクトリ D3	`/D1/D3`

相対パス

カレントディレクトリが D2 であるとき

ファイル F1	`F1`
ファイル F3	`../F3`
ファイル F4	`../../F4`
ディレクトリ D3	`../D3`

カレントディレクトリが D1 であるとき

ファイル F1	`D2/F1`
ファイル F3	`F3`
ファイル F4	`../F4`
ディレクトリ D3	`D3`

Copyright 2005 BohYoh Shibata

第２種【平成８年度秋】午前・問 31

階層構造のディレクトリを用いたファイル管理を行うオペレーティングシステムにおいて、ファイルが置かれているディレクトリを指すときに指定するものはどれか。

ア　拡張子　　イ　サブディレクトリ　　ウ　パス
エ　ルートディレクトリ　　オ　ワイルドカード

第２種【平成７年度秋】午前・問 49

MS-DOS や UNIX におけるディレクトリ構造に関する記述のうち、正しいものはどれか。

ア　サブディレクトリの下に更にサブディレクトリを置くことができる。
イ　ファイルの下にサブディレクトリを置くことができる。
ウ　ファイルをルートディレクトリの下に置くことはできない。
エ　ルートディレクトリの下に更にルートディレクトリを置くことができる。
オ　ルートディレクトリの下にはサブディレクトリを一つ以上置かなければならない。

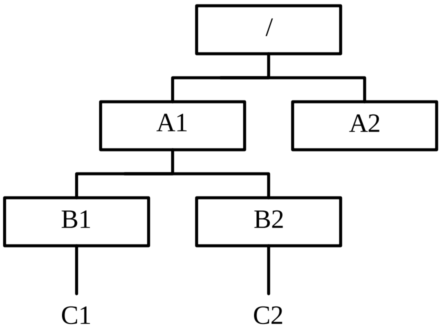
第１種【平成８年度】午前・問 56

UNIX のファイルシステムの特徴として、当てはまる記述はどれか。

ア　ディレクトリを階層化することができない。
イ　ディレクトリをファイルとして操作することができる。
ウ　一つのファイルに複数の名前を付けることはできない。
エ　ファイル操作をする場合には、レコード長を意識する必要がある。
オ　ファイルの所有者以外によるアクセスを制限することはできない。

第２種【平成９年度春】午前・問 46

次の階層ファイルシステムにおいて、カレントディレクトリが B1 であるとき、ファイル C2 を指す相対パスはどれか。ここで、パスの表現において“..”は親ディレクトリを表し、“/”はパス表現の先頭にある場合はルートディレクトリを、中間にある場合は、ディレクトリ名又はファイル名の区切りを表す。また、図中の　　　はディレクトリを表すものとする。

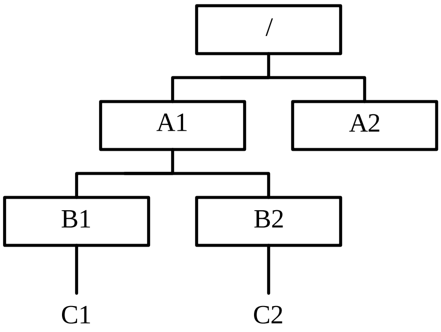


ア　../A1/B2/C2　　イ　../B2/C2
ウ　A1/B2/C2　　エ　/A1/B2/C2

Copyright 2005 BohYoh Shibata

第２種【平成12年度春】午前・問47

図の階層型ファイルシステムにおいて、カレントディレクトリが B1 であるとき、ファイル C2 を指す相対パス名はどれか。パス名の表現において、“..”は親ディレクトリを表し、“ / ”は、パス名の先頭にある場合はルートディレクトリを、中間にある場合はディレクトリ名又はファイル名の区切りを表す。また、図中の　はディレクトリを表すものとする。

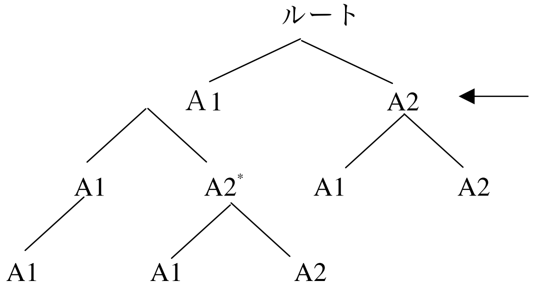


ア　../A1/B2/C2　　イ　../B2/C2
ウ　A1/B2/C2　　エ　/A1/B2/C2

第２種【平成11年度春】午前・問50

複数個のディレクトリ A1、 A2 が図の構造で管理されている。各ディレクトリには、ファイル f が存在する。*印のディレクトリ（カレントディレクトリ）から矢印のディレクトリ配下のファイル f を指定する方法はどれか。

ここで、ファイルの指定方法は、次によるものとする。
ファイルは“ディレクトリ名¥ … ディレクトリ名¥ファイル名”のように、経路上のディレクトリを順に“¥”で区切って指定する。
カレントディレクトリは“.”で表す。
1階層上のディレクトリを“..”で表す。



ア　.¥A2¥f　　イ　..¥..¥A2¥f　　ウ　..¥A1¥..A2¥f　　エ　..¥A2¥f

Copyright 2005 BohYoh Shibata

第２種【平成 13 年度春】午前・問 35

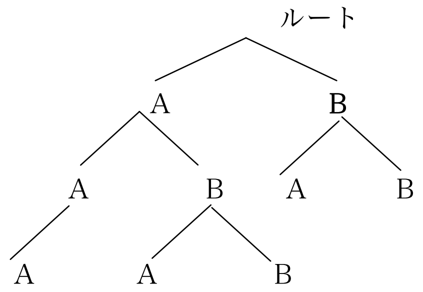
複数個の同名のディレクトリ A1、A2 が、図の構造で管理されている。各ディレクトリには、ファイル f が存在する。 * 印のディレクトリ（カレントディレクトリ）から矢印のディレクトリ配下のファイル f を指定する方法はどれか。
ここで、ファイルの指定方法は、次によるものとする。
ファイルは“ディレクトリ名¥ … ¥ディレクトリ名¥ファイル名”のように、経路上のディレクトリを順に“¥”で区切って指定する。
カレントディレクトリは“.”で表す。
1 階層上のディレクトリを“..”で表す。

ア　.¥A2¥f　　イ　..¥..¥A2¥f　　ウ　..¥A1¥..¥A2¥f　　エ　..¥A2¥f

第２種【平成 12 年度秋】午前・問 46

A、B というディレクトリ名をもつ複数個のディレクトリが図の構造で管理されている。¥A¥B → .. → ..¥B → .¥A のようにカレントディレクトリを移動させた場合、最終的なカレントディレクトリはどこか。ここで、ディレクトリの指定は、次の方法によるものとする。

〔ディレクトリ指定方法〕
(1)“¥”で始まるときは、左端にルートが指定されているものとする。
(2) ディレクトリは“ディレクトリ名 ¥…¥ ディレクトリ名”のように、経路上のディレクトリを上位のディレクトリから“¥”で区切って指定する。
(3) カレントディレクトリは“.”で表す。
(4) 1 階層上のディレクトリを“..”で表す。

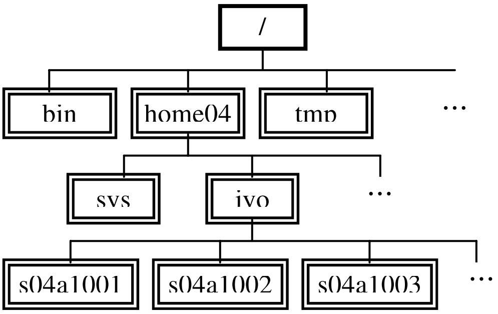


ア　¥A　　イ　¥A¥A　　ウ　¥A¥B¥A　　エ　¥B¥A

Copyright 2005 BohYoh Shibata

■ ホームディレクトリ

マルチユーザの利用を想定した UNIX では、利用者各個人専用のディレクトリが用意されて、そのディレクトリを**ホームディレクトリ**（*home directory*）と呼びます。



2004 年入学の情報工学科の学生のホームディレクトリを絶対パスで表すと、次のようになります。

```
/home04/jyo/s04a1001
/home04/jyo/s04a1002
/home04/jyo/s04a1003
            :
```

なお、ログインした際では、ワーキングディレクトリ（カレントディレクトリ）は、ホームディレクトリとなります。

確　認

ログインしたら、すぐにワーキングディレクトリを表示してみましょう。

```
ipc01%> pwd↲
/home04/jyo/s04a1001
```

ホームディレクトリを表す記号は、~です（「チルダ」と呼びます）。

UNIX では、パラメータを与えずに cd コマンドを実行すると、ワーキングディレクトリは、ホームディレクトリへと移ります。

Copyright 2005 BohYoh Shibata

演　習

ディレクトリ TEST を作ってみましょう（作成済みの場合はエラーとなります)。

```
ipc01%> pwd↵
/home01/jyo/s04a1001
ipc01%> mkdir TEST↵
ipc01%> ls↵
TEST                              ※ ファイルの一覧が表示される
ipc01%>
```

演　習

（続けて）作成したディレクトリの詳細を見てみよう。

```
ipc01%> ls -l↵                    ※ 数字の 1 ではなく小文字のエル
drwxr-xr-x  2 s04a1001     512 Jun 20  10:51 TEST/
drwxr-xr-x  2 s04a1001     512 Jun 20  10:52 WWW/
ipc01%>
```

演　習

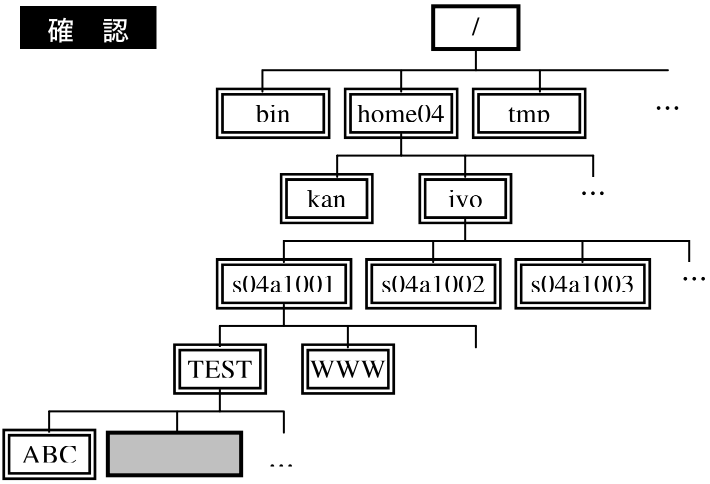
（続けて）TEST ディレクトリの下にも、ABC ディレクトリを作成しましょう。

```
ipc01%> mkdir TEST/ABC↵           ※ TEST/OS は大文字で!!
ipc01%>
```

※ディレクトリの作成を一度行うと、自分で消さない限り残るので、同じ命令を 2 度やったら、エラーが出ることに注意してください。

Copyright 2005 BohYoh Shibata

確　認



現在、ホームディレクトリの下には、少なくとも、以下の二つのサブディレクトリが作られているはずです。

サブディレクトリ **WWW**　　ホームページ用ディレクトリ
サブディレクトリ **TEST**　　OS ディレクトリ

確　認

ホームディレクトリの下の WWW ディレクトリへ移動する方法を確認します。

絶対パスを利用

```
ipc01%> cd /home04/jyo/s04a1001/WWW↵
ipc01%> pwd↵
/home04/jyo/s04a1001/WWW
```

ホームディレクトリからの経路を利用

```
ipc01%> cd ~/WWW↵
ipc01%> pwd↵
/home04/jyo/s04a1001/WWW
```

相対パスを利用

```
ipc01%> cd WWW↵
ipc01%> pwd↵
/home04/jyo/s04a1001/WWW
```

※　この方法は、ワーキングディレクトリがホームディレクトリ以外であれば、無効です。したがって、ホームディレクトリからの経路を利用したほうが確実です。

Copyright 2005 BohYoh Shibata

確　認

ホームディレクトリへ移動後、その下の WWW ディレクトリへ移動。その後、ホームディレクトリの下の OS ディレクトリへ移動する。

例 1

```
ipc01%> cd↲
ipc01%> pwd↲
/home04/jyo/s04a1001/WWW
ipc01%> cd WWW↲
ipc01%> pwd↲
/home03/jyo/s04a1001/WWW
ipc01%> cd ../OS↲
ipc01%> pwd↲
/home04/jyo/s04a1001/OS
```

例 2

```
ipc01%> cd↲
ipc01%> pwd↲
/home04/jyo/s04a1001/WWW
ipc01%> cd ~/WWW↲
ipc01%> pwd↲
/home04/jyo/s04a1001/WWW
ipc01%> cd ~/OS↲
ipc01%> pwd↲
/home04/jyo/s04a1001/OS
```

×例 3 ×

次の例は違うことを行います（WWW ディレクトリの下の OS ディレクトリに移動します）。

```
ipc01%> cd↲
ipc01%> pwd↲
/home04/jyo/s04a1001/WWW
ipc01%> cd WWW↲
ipc01%> pwd↲
/home04/jyo/s04a1001/WWW
ipc01%> cd OS↲
```

Copyright 2005 BohYoh Shibata

確　認

WWWディレクトリの下のOSディレクトリにindex.htmlというファイルを作成しているとする。その内容を表示する。

例１　ホームディレクトリからの経路で表す。

```
ipc01%> cat ~/WWW/OS/index.html↵
```

例２　絶対パスで表す。

```
ipc01%> cat /home04/jyo/s03a1001/WWW/OS/index.html↵
```

例３　相対パスで表す（ワーキングディレクトリがホームディレクトリであるとする）。

```
ipc01%> cat WWW/OS/index.html↵
```

※この方法は、ワーキングディレクトリがホームディレクトリ以外であれば無効。

例４　相対パスで表す（ワーキングディレクトリがWWWディレクトリであるとする）。

```
ipc01%> cat OS/index.html↵
```

※この方法は、ワーキングディレクトリがCディレクトリ以外であれば無効。

例５　相対パスで表す（ワーキングディレクトリがWWWディレクトリの下のOSディレクトリであるとする）。

```
ipc01%> cat index.html↵
```

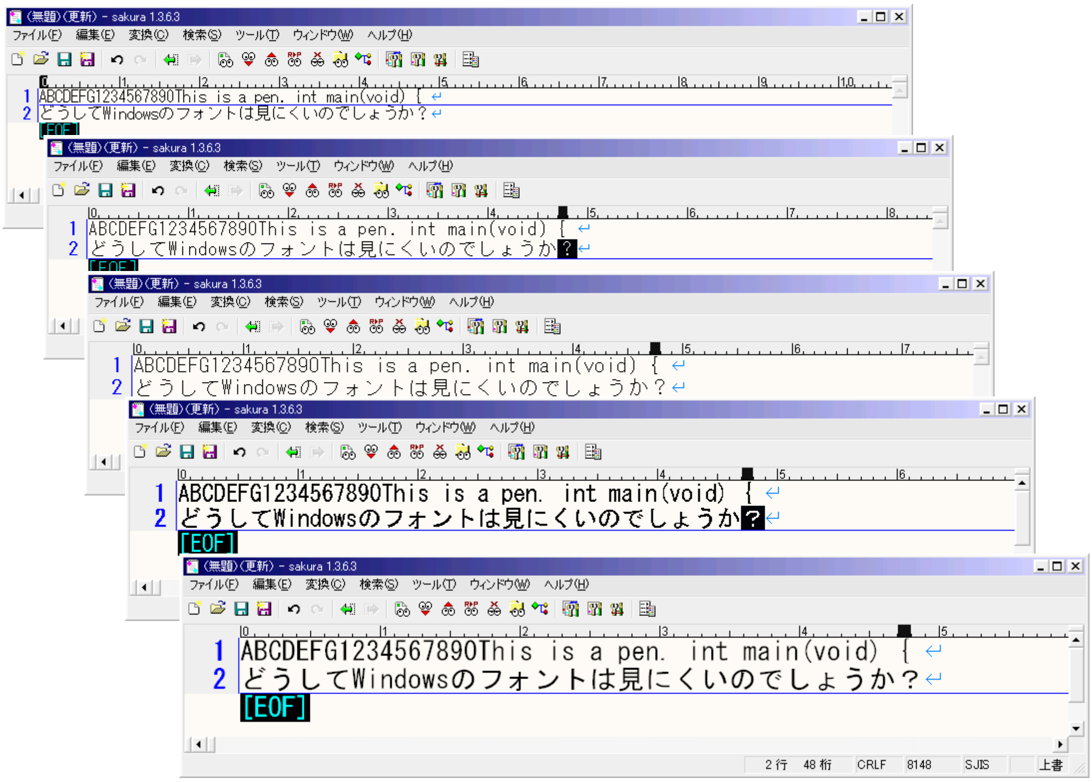
※この方法は、ワーキングディレクトリが~/WWW/OSディレクトリ以外であれば無効。

Copyright 2005 BohYoh Shibata

Windows のフォントについて

ディスプレイは高解像化をたどっていて、携帯電話でも 320×240 の解像度をもつにいたっています。そのため、画面上のフォントが見づらく感じることもあるでしょう。特に、ウェブブラウザ、テキストエディタで、小さくて細い字は耐え難いものです。

MS ゴシックのフォントを、いろいろなサイズで表示したものを以下に示します。上から順に 12、14、16、18、20 ポイントです。



最初の三つは、線が細いですね。四つめは、線が太くなるもののギザギザになります。最後のものは、線が太くなってスムーズになります。これが最も見やすいですね。しかし、これだと**デカ過ぎる**のが最大の欠点です。

あまり小さいフォントに対して、フォントがもつヒント情報を利用したり、アンチエイリアシング（白黒だけで表示せずに、灰色などをもちいてフチをぼかして表示する）をかけたりすると、表示速度が遅くなったり、逆に見づらくなります。

Copyright 2005 BohYoh Shibata

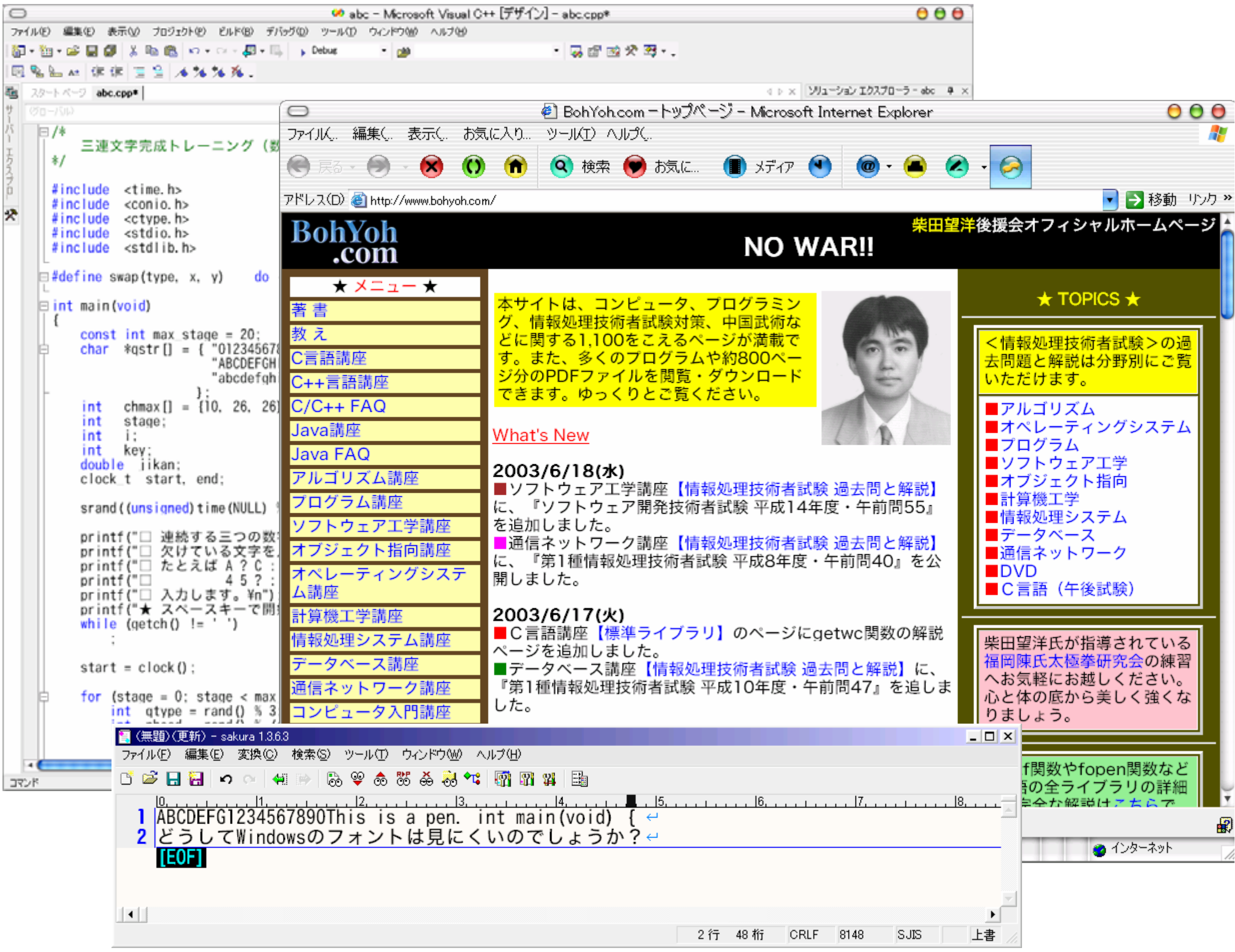
そのため、各フォントファイルに、何ポイントから何ポイントの範囲であれば、ヒント情報を使うのか、あるいはアンチエイリアシング（スムージング）するのかが、埋め込まれています。

MS ゴシックの場合は、16 ポイント以下であれば、本来のフォントとは別に用意された線の細いビットマップフォントが使われます。そして、17～18 ポイントでは、本来のフォント情報が使われるものの、アンチエイリアシングは行われずギザギザです。19 ポイント以上になって、はじめて初めてアンチエイリアシングが行われます。

※ この情報はフォントによって異なります。

私は、大量のフォントを買い込んでいます。さらに一部のフォントファイルは、上記のヒント／スムージング情報を数バイト書きかえることによって、自分にとって見やすい画面を実現しています。

以下のような感じです。



Copyright 2005 BohYoh Shibata

フォントとは何の関係もありませんが、Windows のユーザインタフェース（タイトルバー、ボタンなどを含めてすべて）を自由にカスタマイズするツールがあります。これを愛用しています（前ページの図は Mac の OS X っぽい画面にしたものです）。

■ すべてのフォントでアンチエイリアスがかかるようにする方法

解像度の低いディスプレイでワードを使ってい場合など、明朝体なのかゴシック体なのかの判断するできないことがあります。

すべてのポイントで、アンチエイリアスが効くようにする方法を紹介します。

□TTSDK.EXE をダウンロードします。

日本語フォントの多くは、“MS 明朝”＋“MS P 明朝”といった具合で、複数のフォントから構成されています（.ttc コレクションフォントファイル）。これを個々のファイル(.ttf ファイル)に分割するのが BREAKTTC というツールであり、その逆（.ttf ファイルをまとめて一つの.ttc ファイルを作る）を行うのが MAKETTC というツールです。

Microsoft の TTSDK というツール群に含まれるのですが、現在配布がストップされているようです。以下のサイトからダウンロードします。

http://www.xlsoft.com/jp/products/indigorose/autoplay_readme_02.html

□TTFGASP をダウンロードします。

すべてのポイントでアンチエイリアスが効くようにするツールです。

http://www.geocities.com/SiliconValley/Lakes/8162/

以下に示すのは、“ヒラギノ明朝体 3 & ヒラギノ明朝体 3 等幅”というフォントでの実行例です。

```
// .ttc ファイルを.ttf ファイルへと分割します。
// この例では dshirmn3.ttc から font00.ttf と font01.ttf が作成されます。
        C:¥>breakttc dshirmn3.ttc
        Writing font00.ttf
        Writing font01.ttf
// それぞれのフォントを、アンチエイリアスが効くようにします。
        C:¥>ttfgasp font00.ttf
        C:¥>ttfgasp font01.ttf
// もとの一つのファイルに戻します。
        C:¥>makettc dshirmn3.ttc font00.ttf font01.ttf
```

Copyright 2005 BohYoh Shibata

記 憶 管 理

プログラムのロード

プログラムは、実行されるときに、それを格納した磁気ディスクなどの補助記憶装置から主記憶装置へと読み込まれます。読み込みを行うことを、**ロード**（*load*）するといい、ロードを行うプログラムを**ローダ**（*loader*）と呼びます。

主記憶の空いているところにプログラムをロードできると都合が良いので、通常のプログラムは、任意のアドレスに配置できるようになっており、**再配置可能プログラム**（*relocatable program*）と呼ばれます。

0	プログラム開始
…	
100	200 番地のデータを表示
…	
200	Hello!

1400	
1500	プログラム開始
…	
1600	1700 番地のデータを表示
…	
1700	Hello!

ベースレジスタ方式による再配置では、再配置可能なロードモジュールを主記憶装置にロードし、その先頭番地をベースレジスタに格納します。実行時に、ベースレジスタの値を加算して、実行アドレスを決定します。

なお、プログラムの実行直前に配置するものを**静的再配置**、プログラムの実行中にダイナミックに配置されるものを**動的再配置**と呼ぶ。

第２種【平成８年度秋】午前・問 34

プログラムをロードして実行するとき、プログラムのロード位置に対応してプログラム内のアドレスを補正することを示す用語はどれか。

ア　仮想記憶　　イ　最適化　　ウ　再配置　　エ　翻訳　　オ　連係編集

第２種【平成 12 年度秋】午前・問 35

プログラムをロードして実行するとき、プログラムのロード位置に対応してプログラム内のアドレスを補正することを示す用語はどれか。

ア　仮想記憶　　イ　最適化　　ウ　再配置　　エ　リンキング

Copyright 2005 BohYoh Shibata

動的リンクライブラリ（DLL）

オペレーティングシステムの本体は巨大であり、すべてを主記憶にロードすることはできません。したがって、一部をライブラリの形で用意しておき、必要なときにのみロードできると、記憶域を大幅に節約することができます。

また、オペレーティングシステム上で動作するプログラムから共通して利用するようなライブラリも、それぞれのプログラムで同じものを個別に持つのではなく、必要に応じて主記憶にロードするようにしておくと、各プログラムでライブラリをもつ必要がなくなりますから、実行プログラムの大きさを抑えることもできます。

以上のような目的を実現するのが動的リンクライブラリです。呼び出された時点で、OS がハードディスクなどの補助記憶域装置から主記憶へと読み出してリンクします。

※たとえば MS-Windows であれば、System や System32 フォルダの中に.DLL という拡張子のついたファイルがたくさんあります。これらが動的リンクライブラリです。

第２種【平成９年度春】午前・問 38

動的リンクライブラリ（DLL）に関する正しい記述はどれか。

ア　コンパイル時に、コンパイラによって組み込まれる。

イ　コンパイル前に、プリコンパイラによって組み込まれる。

ウ　実行時に、オペレーティングシステムによって連係される。

エ　ロードモジュール作成時に、連係編集プログラムによって連係され組み込まれる。

Copyright 2005 BohYoh Shibata

第2種【平成11年度秋】午前・問36

動的リンクライブラリ（DLL）に関する記述として、適切なものはどれか。

ア　コンパイル時に、コンパイラによって組み込まれる。
イ　コンパイルの前に、プリコンパイラによって生成される。
ウ　実行時に、OS によって連係される。
エ　ロードモジュール作成時に、連係編集プログラムによって連係され組み込まれる。

第1種【平成12年度】午前・問31

動的リンキングに関する記述として、適切なものはどれか。

ア　仮想記憶方式のコンピュータにおいて、読み込まれたページの論理アドレスを物理アドレスに対応させる。
イ　プログラム実行時に、共用ライブラリやシステムコールライブラリのモジュールをロードする。
ウ　プログラム実行時に、適切なアドレスにロードする。
エ　プログラム実行時に、ヒープ上に可変長データのための領域を獲得する。

第1種【平成8年度】午前・問22

オペレーティングシステムに関する記述のうち、正しいものはどれか。

ア　実行中のプログラムを主記憶装置の異なった領域で実行するために、実行中に新たな絶対アドレスを割り当てる処理を、動的再配置という。
イ　主記憶の利用効率を向上させる一つの方法して、スワッピングがあり、主記憶から外部記憶へプログラムを移すことをスワップイン、その逆をスワップアウトという。
ウ　一つのプログラムのサブモジュールを複数のプロセッサ上で並列に実行することを、マルチプログラミングという。
エ　プログラムの一部をユーザプログラムによって主記憶装置の中の同じ区域にロードし、実行することを、ダイナミックリンケージという。
オ　リンカ（連係編集プログラム）が出力し、かつオペレーティングシステムのもとで実行可能なプログラムを、オブジェクトモジュールという。

第1種【平成8年度】午前・問27

再配置可能なロードモジュールのロードと実行に、最も関係の深いレジスタはどれか。

ア　インデックスレジスタ　　イ　シフトレジスタ　　ウ　汎用レジスタ
エ　ベースレジスタ　　オ　命令レジスタ

Copyright 2005 BohYoh Shibata

再入可能（リエントラント）

実行中であっても、他のタスクから呼び出すことのできるプログラムは、**リエントラント（再入可能）**であるといわれます。

再入可能プログラムを実現するためには、プログラムを手続き部分とデータ部分に分割して、プロセスごとにデータ部分をスタックとしてもつ必要があります。

※　エントランスは≪入り口≫という意味ですね。英語の re は、再びという意味です（リプレイ＝再生／リコンストラクション＝再構築）。

第２種【平成 10 年度春】午前・問 40

処理が終了していないプログラムが、別のプログラムから再度呼び出されることがある。このプログラムが正しく実行されるために備えるべき性質はどれか。

ア　再帰的（リカーシブ）　　イ　再入可能（リエントラント）
ウ　再配置可能（リロケータブル）　　エ　再利用可能（リユーザブル）

第２種【平成 12 年度秋】午前・問 36

処理が終了していないプログラムが、別のプログラムから再度呼び出されることがある。このプログラムが正しく実行されるために備えるべき性質はどれか。

ア　再帰的（リカーシブ）　　イ　再使用可能（リユーザブル）
ウ　再入可能（リエントラント）　　エ　再配置可能（リロケータブル）

第２種【平成８年度春】午前・問 34

次の記述中の［　］に入れるべき適切な字句の組合せはどれか。

主記憶にロードされた一つの［ a ］を複数の［ b ］で共用して並行に実行するためには、その［ a ］が［ c ］である必要がある。

	a	b	c
ア	サブルーチン	タスク	再帰的
イ	サブルーチン	プログラム	再帰的
ウ	タスク	サブルーチン	再帰的
エ	タスク	プログラム	再入可能
オ	プログラム	タスク	再入可能

Copyright 2005 BohYoh Shibata

第２種【平成７年度秋】午前･問 37

再入可能（リエントラント）プログラムの説明として、最も適切なものはどれか。

ア　一度実行した後、再ロードすることなく再び実行を繰り返しても、正しく実行が可能である。
イ　実行中であっても、重ねて同一プログラムの実行が可能である。
ウ　主記憶上のどこのアドレスに配置しても実行が可能である。
エ　障害によって実行が中断した場合、チェックポイントにおいて記録したデータを使用して実行の再開が可能である。
オ　複数のセグメントに分割し、セグメント単位にローディングして実行が可能である。

第２種【平成９年度春】午前･問 35

再入可能（リエントラント）プログラムの説明として、最も適切なものはどれか。

ア　一度実行した後、ロードし直さずに再び実行を繰り返しても、正しい結果が得られる。
イ　主記憶上のどこのアドレスに配置しても実行することが可能である。
ウ　同時に複数のタスクが共有して実行しても、正しい結果が得られる。
エ　複数のセグメントに分割されており、セグメント単位にローディングして実行することが可能である。

第２種【平成 12 年度春】午前･問 38

再入可能プログラムの特徴として、適切なものはどれか。

ア　実行時に必要な手続きを補助記憶装置から呼び出しながら動作する。実行時の主記憶領域の大きさに制限があるときに、有効な手法である。
イ　手続きの内部から自分自身を呼び出すことができる。
ウ　複数のタスクで利用するとき、一時に一つのタスクでの使用が可能である。
エ　複数のタスクによって並行して実行されても、それぞれのタスクに正しい結果を返す。

第２種【平成 11 年度春】午前･問 35

再入可能（リエントラント）プログラムの説明として、最も適切なものはどれか。

ア　一度実行した後、ロードし直さずに再び実行を繰り返しても、正しい結果が得られる。
イ　実記憶上のどこのアドレスに配置しても実行することが可能である。
ウ　複数のセグメントに分割されており、セグメント単位にロードして実行することが可能である。
エ　複数のタスクで並行して実行しても、正しい結果が得られる。

Copyright 2005 BohYoh Shibata

第1種【平成8年度】午前・問30

再入可能（リエントラント）プログラムに関する記述のうち、正しいものはどれか。

ア　再入可能プログラムでは、変数をタスク単位に格納しなければならない。
イ　再入可能プログラムは、C言語では作成できない。
ウ　再入可能プログラムは、逐次使用可能プログラムから呼び出すことはできない。
エ　実行途中で待ち状態が発生するプログラムは、再入可能ではない。
オ　逐次再使用可能なプログラムは、再入可能プログラムとして使用できる。

第1種【平成10年度】午前・問28

再入可能（リエントラント）プログラムに関する記述のうち、正しいものはどれか。

ア　再入可能プログラムは、逐次再使用可能プログラムから呼び出すことはできない。
イ　再入可能プログラムは、局所変数をタスク単位に格納しなければならない。
ウ　実行途中で待ち状態が発生するプログラムは、再入可能ではない。
エ　逐次再使用可能なプログラムは、再入可能プログラムとして使用できる。

第1種【平成9年度】午前・問25

ある手続きが複数のプロセスによって並行して実行されるとき、それぞれのプロセスに正しい結果を返すことができる場合、この手続きの性質を何というか。

ア　再帰的　　イ　再入可能　　ウ　再配置可能　　エ　逐次再使用可能

第1種【平成11年度】午前・問28

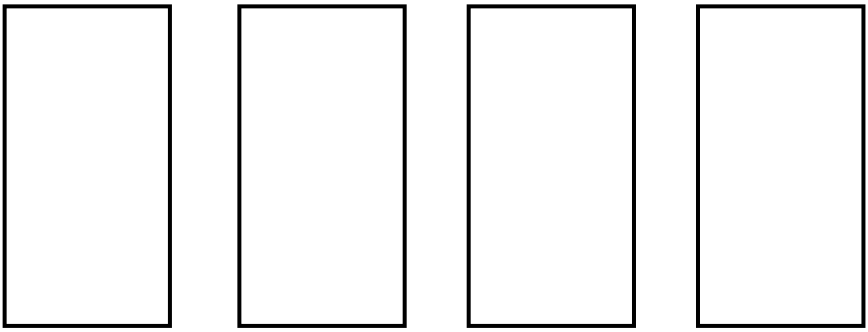
リアルタイムシステムにおいて、複数のタスクから並行して呼び出される共用ライブラリのプログラムに要求される性質として、適切なものはどれか。

ア　リエントラント　　イ　リカーシブ　　ウ　リユーザブル　　エ　リロケータブル

Copyright 2005 BohYoh Shibata

フラグメンテーション

オペレーティングシステムが、記憶領域の割当てと解放を繰り返すことによって、こま切れの未使用領域が多数できてしまうことを**フラグメンテーション**という。



第２種【平成９年度春】午前・問 34

オペレーティングシステムが、記憶領域の割当てと解放を繰り返すことによって、こま切れの未使用領域が多数できてしまう場合がある。この現象を何というか。

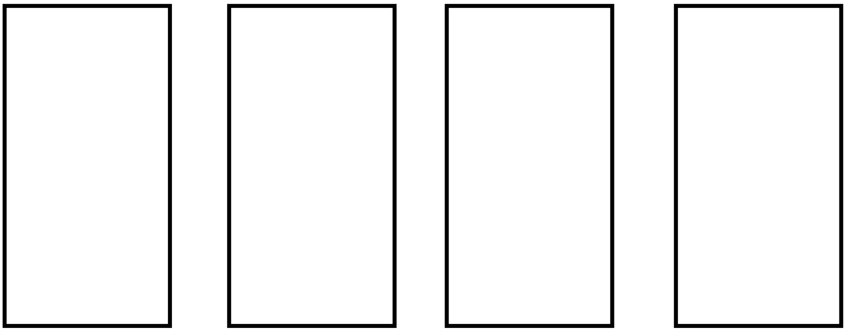
ア　コンパクション　　イ　スワッピング　　ウ　フラグメンテーション
エ　ページング

第２種【平成 12 年度春】午前・問 36

OS が記憶領域の割当てと解放を繰り返すことによって、こま切れの未使用領域が多数できてしまう場合がある。この現象を何というか。

ア　コンパクション　　イ　スワッピング
ウ　フラグメンテーション　　エ　ページング

フラグメンテーションによる空き領域を“詰める”ことを、**コンパクション**と呼ぶ。



Copyright 2005 BohYoh Shibata

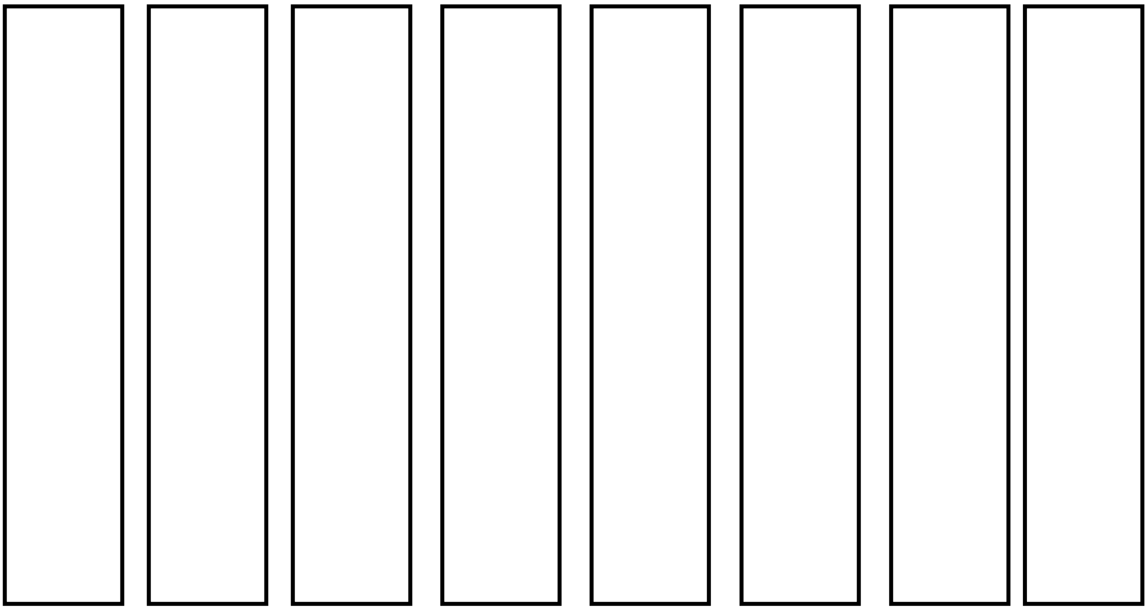
オーバレイ

ロードモジュールを、同時に実行されることのない複数個の部分（セグメント）に分割し、実行時にセグメントを交互にローディングする方式を、**オーバレイ**と呼ぶ。

```
abc() { /*・・・*/ }
xy()   { /*・・・*/ }
z()     { /*・・・*/ }

xyz()
{
        xy();
        z();
}

int main(void)
{
        abc();
        xyz();
}
```



Copyright 2005 BohYoh Shibata

第２種【平成７年度春】午前・問 31

ロードモジュールを、同時に実行されることのない複数個の部分（セグメント）に分割し、実行時にセグメントを交互にローディングする方式はどれか。

ア　オーバレイ　　イ　スワッピング　　ウ　ダイナミックリンク
エ　動的再配置　　オ　ロールインロールアウト

第２種【平成 12 年度春】午前・問 37

ロードモジュールを、排他的に実行される複数個の部分（セグメント）に分割し、実行時に必要なセグメントをローディングする方式はどれか。

ア　オーバレイ　　イ　スワッピング
ウ　動的再配置　　エ　動的リンキング

Copyright 2005 BohYoh Shibata

ガーベジコレクション

プログラムが使用しなくなったヒープ領域を回収して再度使用可能にする処理が、**ガーベジコレクション**（*garbage collection*）です。

第1種【平成10年度】午前・問27

プログラム実行時のメモリ管理に関する記述として、正しいものはどれか。

ア　主記憶域の容量をこえるプログラムを実行できるようにすることを、メモリコンパクションという。
イ　プログラムが使用しなくなったヒープ領域を回収して再度使用可能にする処理を、ガーベジコレクションという。
ウ　プログラム実行中必要になった時点で、ライブラリに用意されているプログラムモジュールをロードする方法を、動的再配置という。
エ　メモリを有効に利用するために、オブジェクトモジュールを主記憶上で移動させ、プログラムを実行することを、動的リンキングという。

Copyright 2005 BohYoh Shibata

■ 仮想記憶方式

一般に、主記憶装置の容量は、ユーザが実行する全てのプログラムを一度に格納するには十分ではありません。補助記憶装置などを用いて、実際の主記憶装置の容量以上のアドレス空間を仮想的に作り出す**仮想記憶**（*virtual storage*）を採用します。

■ ページング

仮想記憶をページという単位で格納し、実記憶をページ枠で管理する方式が**ページング**です。**実記憶領域の利用効率が高く、領域管理も容易です**。

主記憶装置をテーブルと考えましょう。このテーブルには、10枚のカードしか置けないとします。しかし、テーブルの横に大きな補助テーブルを用意し、テーブルと補助テーブル上のトランプを、うまく入れかえることによって、10枚以上のカードをテーブル上で扱えるようになります。

カードの入れかえの作業、すなわち、あるプログラムの実行中に，それより優先度の高いプログラムを実行するために、もとのプログラムを補助記憶装置と主記憶とで交換する作業が**スワッピング**です。実行中のジョブを補助記憶装置に退避するのが**スワップアウト**、優先度の高いジョブを補助記憶装置から主記憶装置に読み込むのが**スワップイン**です。

MS-Windowsを使っていて、（特に主記憶が少ないパソコンで）やたらとディスクアクセスが起こるのは、このためです。

なお、スワッピングを行う補助記憶装置は、高速かつ大容量でなければなりませんから、通常は**ハードディスク**が使われます。

仮想記憶上のプログラムは、仮想アドレス（論理アドレス）、実記憶上では実アドレス（物理アドレス）で格納されます。これを、命令を実行するとき、あるいは、データを参照するときに、一つ一つ仮想アドレスから実アドレスにページテーブルを使って変換します。これが、**動的アドレス変換**です。

プログラムの実行時に**ページ表**あるいは**セグメント表**を使って、論理アドレスを物理アドレスに変換します。

※　普通の大学ノートは、順番に書いていかないといけませんが、ルーズリーフだったら、ページの挿入・削除が気軽に行えますね。

動的アドレス変換

仮想アドレス　→　ページテーブル　→　実アドレス

Copyright 2005 BohYoh Shibata

実記憶上にないページをアクセスした場合の処理と状態の順番は、次のようになります。
ページフォールト → 置換え対象ページの決定 → ページアウト → ページイン

プロセスの多重度を上げると、実メモリの要求が増えるため、ページフォールトとの発生が増加します。
ページング方式を用いて仮想記憶を実現しているシステムにおいて、実メモリが小さく、ページングの回数が多くなるとページフォールトが頻発して、**スラッシング**が発生します。そうすると、主記憶と補助記憶との間のページ転送量が多くなって、CPU の利用効率は低くなり本来の処理ができなくなります。

■ ページ入替方式

・FIFO（First-In First-Out）
実記憶域に最初にページインしたページを最初にページアウトします。すなわち、もっとも長い間実記憶域に存在していたページをページアウトする方式です。

・LRU（Least Recently Used）
最後に使われてから最も時間が経過しているページを最初にページアウトします。すなわち、最も長い間参照されていないページをページアウトする方式です。

なお、ある時点から一定時間前にプログラムが参照したページの集合を**ワーキングセット**といいます。

■ ページサイズ

ページサイズを小さくすると、無駄な領域が少なくなって資源の利用効率は向上しますが、アドレス変換の頻度は増大します。
ページサイズを大きくすると、無駄な領域が多くなってスワッピングの頻度が増大します。

Copyright 2005 BohYoh Shibata

第２種【平成８年度秋】午前･問23

仮想記憶方式を用いているシステムにおいて、実記憶に格納しきれないプログラムやデータを格納するための装置として最も適切なものはどれか。

ア CD-ROM　イ VRAM　ウ キャッシュメモリ
エ ハードディスク　オ フロッピーディスク

第２種【平成10年度秋】午前･問39

動的アドレス変換の説明として、最も適切なものはどれか。

ア　仮想記憶システムにおいて、仮想アドレスから実アドレスへの変換を行うこと。
イ　実行中のプログラムを移動して新しい場所で実行できるように、プログラムの基底アドレスを変更すること。
ウ　主記憶に対する読み書きを、キャッシュメモリで代行すること。
エ　プログラムの実行途中にモジュールを追加するため、モジュール間のアドレス参照を解決すること。

第２種【平成８年度秋】午前･問35

仮想記憶方式で用いられる記憶管理の手法又は機構を二つ選べ。

ア　オーバレイ　イ　スプーリング　ウ　動的アドレス変換
エ　動的リンキング　オ　ページング

第２種【平成７年度秋】午前･問38

仮想記憶方式の記憶管理に関係の深いものを二つ選べ。

ア　キャッシュ　イ　セグメント　ウ　パイプライン　エ　ページ　オ　リフレッシュ

第２種【平成10年度秋】午前･問36

あるプログラムの実行中に、それより優先度の高いプログラムを実行するために、もとのプログラムを補助記憶装置に移して優先度の高いプログラムをロードする処理はどれか。

ア　オーバレイ　イ　スワッピング　ウ　ページング　エ　リロケーション

Copyright 2005 BohYoh Shibata

第２種【平成６年度秋】午前・問 27

オペレーティングシステムの機能とそれに関連する用語の組合せで、[ ]に入れるべき適切な字句はどれか。

ジョブ管理	－	JCL
[ ]	－	ページング
ファイル管理	－	ディレクトリ
入出力管理	－	バッファリング
運用管理	－	システムモニタリング

ア　記憶管理　　イ　システム管理　　ウ　装置管理　　エ　通信管理　　オ　プロセス管理

第２種【平成 10 年度春】午前・問 34

ページング方式に関する記述はどれか。

ア　実記憶空間と仮想記憶空間を、固定長の領域に区切り、対応づけて管理する方式
イ　主記憶装置の異なった領域で実行できるように、プログラムを再配置する方式
ウ　主記憶を、同時に並行して読み書き可能な複数の領域に分ける方式
エ　ファイル記憶媒体への読み書きをするとき、複数のレコードをまとめて行う方式

基本【平成 13 年度春】午前・問 30

仮想記憶におけるセグメンテーション方式とページング方式に関する記述のうち、ページング方式の特徴はどれか。

ア　仮想アドレス空間の管理単位である領域の大きさを、実行時に動的に変更できる。
イ　実記憶領域の利用効率が高く、領域管理も容易である。
ウ　プログラムからみた論理的な単位でアクセス保護を行うことができる。
エ　プログラム実行中のモジュールの取込みや共有を容易に行うことができる。

第２種【平成８年度春】午前・問 33

記憶空間を一定の大きさに区切って管理し、仮想記憶を実現する方式を何というか。

ア　スラッシング　　イ　スワッピング　　ウ　ブロッキング
エ　ページング　　オ　ローディング

Copyright 2005 BohYoh Shibata

第２種【平成９年度秋】午前・問 33

仮想記憶方式の一つに、仮想アドレス空間を固定長の領域に分割して管理するものがある。この固定長の領域を示す用語はどれか。

ア　セクタ　　イ　フレーム　　ウ　ページ　　エ　モジュール

第２種【平成 11 年度秋】午前・問 31

仮想記憶方式の一つに、仮想アドレス空間を固定長の領域に分割して管理するものがある。この固定長の領域を示す用語はどれか。

ア　セクタ　　イ　セグメント　　ウ　フレーム　　エ　ページ

第２種【平成 12 年度春】午前・問 32

ページング方式の仮想記憶において、実記憶上にないページをアクセスした場合の処理と状態の順番として、適切なものはどれか。ここで、実記憶には現在、空きページはないものとする。

ア　置換え対象ページの決定 → ページアウト → ページフォールト → ページイン
イ　置換え対象ページの決定 → ページイン → ページフォールト → ページアウト
ウ　ページフォールト → 置換え対象ページの決定 → ページアウト → ページイン
エ　ページフォールト → 置換え対象ページの決定 → ページイン → ページアウト

第２種【平成 10 年度春】午前・問 35

ページング方式を用いて仮想記憶を実現しているシステムにおいて、スラッシングが発生しているときの状況はどれか。

ア　CPU の利用効率は高く、主記憶と補助記憶との間のページ転送量は多い。
イ　CPU の利用効率は高く、主記憶と補助記憶との間のページ転送量は少ない。
ウ　CPU の利用効率は低く、主記憶と補助記憶との間のページ転送量は多い。
エ　CPU の利用効率は低く、主記憶と補助記憶との間のページ転送量は少ない。

第２種【平成９年度春】午前・問 19

仮想記憶のページ入替方式のうち、使われたのが最も古いページを入れ替えるものはどれか。

ア　FIFO　　イ　LFU　　ウ　LIFO　　エ　LRU

Copyright 2005 BohYoh Shibata

第２種【平成 10 年度秋】午前･問 37

仮想記憶におけるページ置換えアルゴリズムの一つである LRU を説明した記述はどれか。

ア　あらかじめ設定されている優先度が最も低いページを追い出す。
イ　主記憶に存在している時間が最も長いページを追い出す。
ウ　主記憶に存在している時間が最も短いページを追い出す。
エ　最も長い間参照されていないページを追い出す

第２種【平成 12 年度秋】午前･問 28

仮想記憶管理におけるページ置換えに関する記述のうち、LRU 制御方式はどれか。

ア　各ページに参照フラグと変更フラグを付加して管理し、参照なし、変更なしのページを優先して置き換える。
イ　主記憶にあるすべてのページを同一の確率でランダムに選択し、置き換える。
ウ　最も長い間参照されていないページを置き換える。
エ　最も長い間主記憶にあったページを置き換える。

基本【平成 13 年度春】午前･問 31

ページ置換えアルゴリズムにおける LRU 方式の説明として、適切なものはどれか。

ア　一番古くから存在するページを置き換える方式
イ　最後に参照されたページを置き換える方式
ウ　最後に参照されてからの経過時間が最も長いページを置き換える方式
エ　参照回数の最も少ないページを置き換える方式

第２種【平成 11 年度春】午前･問 21

仮想記憶に関する記述のうち，正しいものはどれか。

ア　16 ビットのマシンでは，アドレス空間が小さいので仮想記憶が必要になるが、32 ビットのマシンでは、アドレス空間が大きいので仮想記憶は必要ない。
イ　仮想記憶を採用した場合，プログラム実行時のアドレスを前もって決めることができないので、動的リンキング機能を使ってプログラムを実行する。
ウ　実装された主記憶の容量を越えるアドレス空間の部分を補助記憶に割り付ける。
エ　プログラムの実行時にページ表（又はセグメント表）を使って、論理アドレスを物理アドレスに変換する。

Copyright 2005 BohYoh Shibata

第２種【平成６年度秋】午前・問 16

仮想記憶に関する記述のうち、正しいものはどれか。

ア　仮想記憶空間は、主記憶から補助記憶へ連続したアドレスが割り付けられる。
イ　個々のプログラムに対しては、主記憶の容量を超えない範囲で仮想空間が与えられる。
ウ　主記憶に置かれていないプログラムは、キャッシュメモリ上で実行される。
エ　ページング方式は、仮想記憶を実現する一方法である。
オ　マルチタスクの実現には、仮想記憶が必要である。

第２種【平成 12 年度秋】午前・問 31

性能見積りを行う場合のメモリ使用率の算出に関する説明として、適切なものはどれか。

ア　OS、ユーザ空間ごとに、必要なメモリの総和と実搭載量から算出する。
イ　加重平均ダイナミックステップ数とシステム全体の処理データ量から算出する。
ウ　処理モデルごとに、すべての回線を流れる電文の電文長と通信量から算出する。
エ　処理モデルごとに、入出力レコードのサイズとアクセス回数から算出する。

第１種【平成８年度】午前・問 24

ページング方式を用いた仮想記憶システムの主記憶管理方式に関係の深い用語を二つ選べ。

ア　オーバレイ　　イ　スラッシング　　ウ　メモリコンパクション
エ　メモリフラグメンテーション　　オ　ワーキングセット

第１種【平成 12 年度】午前・問 28

主記憶装置をページに分割し、それぞれのページに R（読込み許可）、W（書込み許可）、E（実行許可）という属性を与えて、アクセスを制限しているシステムがある。このシステムにおいて、プログラムを実行する場合、コード領域、データ領域がロードされるページに付与される属性の組合せとして、最も適切なものはどれか。

	コード領域	データ領域
ア	ERW	ERW
イ	ER	RW
ウ	RW	RW
エ	R	W

Copyright 2005 BohYoh Shibata

第 1 種【平成 8 年度】午前・問 25

仮想記憶管理におけるページ置換えアルゴリズムとして、LRU 方式を採用する。参照かつ更新されるページ番号の順番が、2→3→5→8→2→3→6→2→3→5→1→6 で、主記憶のページ枠が 4 ページ分のとき、ページアウトは何回発生するか。ここで、初期状態では、主記憶ページには何も読み込まれていないものとする。

ア　2　　イ　3　　ウ　4　　エ　5　　オ　6

第 1 種【平成 12 年度】午前・問 23

仮想記憶管理におけるページ置換えアルゴリズムとして、LRU 方式を採用する。参照かつ更新されるページ番号の順番が、2→3→5→8→2→3→6→2→3→5→1→6 で、実記憶のページ枠が 4 のとき、ページフォールトに伴って発生するページアウトは何回か。ここで、初期状態では、実記憶にはいずれのページも読み込まれていないものとする。

ア　3　　イ　4　　ウ　5　　エ　6

第 1 種【平成 10 年度】午前・問 26

仮想記憶管理のページ置換えアルゴリズムとして FIFO を用いる。実記憶のページ枠が 3 ページ分で、参照する仮想ページの番号の順が、4, 3 ,2 ,1, 4, 3, 5, 4, 3, 2, 1, 5 のとき、ページインは何回行われるか。ここで、初期状態では、実記憶には何も読み込まれていないものとする。

ア　7　　イ　8　　ウ　9　　エ　10

Copyright 2005 BohYoh Shibata

第 1 種【平成 11 年度】午前・問 96

仮想メモリに関する記述のうち、正しいものはどれか。

ア　実メモリに仮想ページが存在しないことをミスヒットと呼ぶ。
イ　ページサイズを大きくすると、スワップが発生しやすくなる。
ウ　ページサイズを小さくすると、スラッシングが発生しやすくなる。
エ　マルチタスキングでページフォールトを減らすには、プロセスの多重度をあげる。

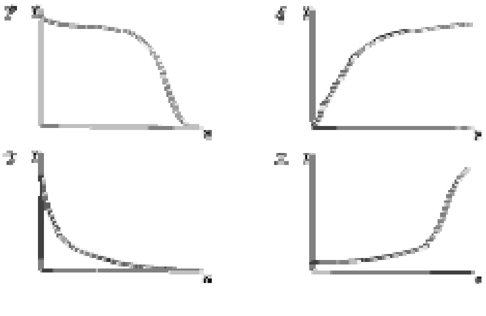
ソフトウェア【平成 13 年度】午前・問 28

システム障害の原因となるメモリリークに関する記述として、適切なものはどれか。

ア　高速化のためにメモリ上に常駐化させていたプログラムの一部が、勝手にページアウトされてしまう。
イ　自プロセスが獲得したメモリの領域外に書き込んでしまい、ほかのプロセスの領域を破壊してしまう。
ウ　多数のプロセスが開始・終了を繰り返すうちに、数多くの未使用領域が断片的に発生し、メモリの使用効率が低下する。
エ　プロセスの獲得したメモリが、不要になった後も解放されず、システムで使用できるメモリが徐々に減少する。

第 1 種【平成 11 年度】午前・問 22

仮想記憶方式における、割当て主記憶容量（x）とページフォールトの発生率（y）の関係はどれか。ここで、ページングアルゴリズムは LRU 方式を採用し、ページ置換えはランダムに発生するものとする。



第１種【平成11年度】午前・問27

記憶管理機能の記述に関して、a～cに対応する機能の正しい組合せはどれか。

機能名	特　徴
a	あらかじめプログラムを幾つかの単位に分けて二次記憶装置に格納しておき、プログラムの指定に基づいて実記憶装置との間で出し入れをする。
b	主記憶とプログラムを固定長の単位に分割し、効率よく記憶管理する。このため、少ない実記憶で大きなプログラムの走行を可能にしている。
c	プログラムを一時的に停止させ、使用中の実記憶の内容を二次記憶装置に退避する。再開時には、退避した内容を再ロードし、元の主記憶状態に戻す。

	a	b	c
ア	オーバレイ機能	ページング機能	スワッピング機能
イ	スワッピング機能	オーバレイ機能	ページング機能
ウ	スワッピング機能	ページング機能	オーバレイ機能
エ	ページング機能	オーバレイ機能	スワッピング機能

ソフトウェア【平成13年度】午前・問29

記憶管理機能の記述に関して、a～cに対応する機能の正しい組合せはどれか。

機能名	特　徴
a	あらかじめプログラムを幾つかの単位に分けて補助記憶装置に格納しておき、プログラムの指定に基づいて実記憶装置との間で出し入れをする。
b	主記憶とプログラムを固定長の単位に分割し、効率よく記憶管理する。このため、小さい主記憶装置で大きなプログラムの実行を可能にしている。
c	プログラムを一時的に停止させ、使用中の主記憶装置の内容を補助記憶装置に退避する。再開時には、退避した内容を主記憶装置に再ロードし、元の状態に戻す。

	a	b	c
ア	オーバレイ	ページング	スワッピング
イ	スワッピング	オーバレイ	ページング
ウ	スワッピング	ページング	オーバレイ
エ	ページング	オーバレイ	スワッピング

Copyright 2005 BohYoh Shibata

データ管理

オペレーティングシステムのデータ管理機能は、各種の補助記憶装置へのアクセス手段を、装置に依存しないインタフェースで提供します。

■ アクセス方式

さまざまな情報が記録されているファイルの中からデータを取り出す順番を指定する方法がアクセス方式です。データが記憶されている補助記憶装置の種類に応じて、適切なアクセス方式を選択する必要があります。

順次アクセス、直接アクセス、動的アクセス

■ ファイル編成方式

アクセス方式と密接な関係があります。

順編成ファイル、直接編成ファイル、索引編成ファイル、区分編成ファイル etc…

■ カタログ管理

補助記憶装置に記憶されている膨大な量のファイルの中から、必要なファイルを素早く正確に見つけるための機能で、

・データが記録されている装置
・データが記録されているボリューム
・ボリューム内のファイルの記録場所

などの情報を、カタログと呼ばれる場所に記録・管理することです。

マスタカタログ

オペレーティングシステム自身が使用するファイルの情報を記録する領域。

ユーザカタログ

利用者が使用するファイルの情報を記録する領域。

第２種【平成10年度秋】午前･問38

オペレーティングシステムのデータ管理のもつ役割として、正しいものはどれか。

ア　各種の入出力装置を、正確かつ効率よく動作させるように制御する。
イ　各種の補助記憶装置へのアクセス手段を、装置に依存しないインタフェースで提供する。
ウ　タスクの状態管理を行い、処理装置を有効に利用する。
エ　補助記憶装置を利用して、大きな論理記憶空間を提供する。

Copyright 2005 BohYoh Shibata

入出力管理

複数のジョブを同時に処理する多重プログラミングでは、処理装置に対して同時に入出力要求が発生することがある。この問題を解決するのが入出力管理の働きです。

実記憶と入力装置や出力装置の設置されたチャネルが、オペレーティングシステムの入出力監視プログラム中のチャネルプログラムの指示によって、入出力管理を行います。

チャネルには、

・セレクタチャネル

・バイトマルチプレクサチャネル

・ブロックマルチプレクスチャネル

があります。いずれのチャネルも処理装置と同じような機能をもち、処理装置から入出力の指令が出ると、入出力監視プログラムにあらかじめ用意されているチャネルプログラムがチャネルを動作させるようになっています。チャネルプログラムの指令によってチャネルが入出力管理を行うため、処理装置は入出力管理から解放され、別の処理を行うことができます。

プログラム制御方式

処理装置のレジスタを経由して、主記憶装置と入出力装置の間でデータ転送を行う方式である。

DMA 制御方式

DMA は、direct memory access の略です。処理装置を介さずに、システムバスなどに接続されたデータ転送専用のハードウェアによって、主記憶装置と入出力装置の間で直接転送を行う方式。

チャネル制御方式

DMA 制御方式の一方式であり、入出力専用のハードウェアがデータ転送制御のためのプログラムを主記憶から自律的に読み出して入出力装置を制御することによって、並行処理の度合いを高めることができる。

スプーリング

周辺装置の動作を処理装置の動作と独立に並行して行わせるのが**スプーリング**です。プリンタなどの低速装置への出力データをいったん高速な磁気ディスクに格納しておき、その後に出力データを目的の装置に出力します。

ちなみに、SPOOL は、Simultaneous Peripheral Operation Online の略です。

Copyright 2005 BohYoh Shibata

バッファプール

プログラムの入出力と処理の並行動作を高めることによって性能向上を図る。このために、主記憶上に入出力を行うための領域を多数用意し、複数のプログラムで共用します。

第 2 種【平成 7 年度春】午前・問 32

スプーリングに関して、最も適切な記述はどれか。

ア　相手装置や通信ネットワークに関係なく、統一的な通信手順を提供する。
イ　外部記憶装置を用いて、主記憶装置より大きな仮想記憶を提供する。
ウ　コンピュータシステムの運転経過の情報を記録する。
エ　周辺装置の動作を処理装置の動作と独立に並行して行わせる。
オ　物理レコードを意識することなく、論理レコード単位での処理を可能とする。

第 2 種【平成 11 年度春】午前・問 34

スプーリングの説明として，適切なものはどれか。

ア　キーボードなどからの入力データを主記憶上のキューにいったん保存しておく。
イ　通信データを直接通信相手の装置に送らず、あらかじめ登録しておいた代理の装置に送る。
ウ　二つ以上のプログラム（タスク、プロセス）に、時間で区切って CPU を割り当て、並行して実行する。
エ　プリンタなどの低速装置への出力データをいったん高速な磁気ディスクに格納しておき、その後に出力データを目的の装置に出力する。

第 2 種【平成 10 年度春】午前・問 36

スループットを高めるため、主記憶装置と低速の入出力装置とのデータ転送を、高速の補助記憶装置を介して行う方式はどれか。

ア　スプーリング　　イ　スワッピング　　ウ　ブロッキング　　エ　ポーリング

第 1 種【平成 12 年度】午前・問 27

システム全体のスループットを高めるため、主記憶装置と低速の入出力装置とのデータ転送を、高速の補助記憶装置を介して行う方式はどれか。

ア　スプーリング　　イ　スワッピング　　ウ　ブロッキング　　エ　ポーリング

Copyright 2005 BohYoh Shibata

第１種【平成12年度】午前・問19

入出力制御方式に関するａ～ｃの記述と用語の適切な組合せはどれか。

a　プロセッサのレジスタを経由して、主記憶装置と入出力装置の間でデータ転送を行う方式である。

b　プロセッサを介さずに、システムバスなどに接続されたデータ転送専用のハードウェアによって、主記憶装置と入出力装置の間で直接転送を行う方式である。

c　ｂの一方式であり、入出力専用のハードウェアがデータ転送制御のためのプログラムを主記憶から自律的に読み出して入出力装置を制御することによって、並行処理の度合いを高めることができる。

	a	b	c
ア	パイプライン制御方式	DMA 制御方式	チャネル制御方式
イ	パイプライン制御方式	チャネル制御方式	DMA 制御方式
ウ	プログラム制御方式	DMA 制御方式	チャネル制御方式
エ	プログラム制御方式	チャネル制御方式	DMA 制御方式

ソフトウェア【平成13年度】午前・問27

入出力管理機能の記述に関して、ａ～ｃに対応する機能の正しい組合せはどれか。

機能	特　徴
a	入出力装置に依存した処理を行い、装置の種類ごとに用意され、1台又は複数台の装置を制御する。読出し、書込みなどの入出力要求が出されると、その装置を直接操作・管理する。
b	ファイルのプリンタ出力やシリアル回線を介したファイル転送のように、それほど急を要さない入出力は、専用のプロセスに依頼して、入出力動作とプログラムの実行を並行して行う。
c	プログラムの入出力と処理の並行動作を高めることによって性能向上を図る。このために、主記憶上に入出力を行うための領域を多数用意し、複数のプログラムで共用する。

	a	b	c
ア	スプーリング	デバイスドライバ	バッファプール
イ	スプーリング	バッファプール	デバイスドライバ
ウ	デバイスドライバ	スプーリング	バッファプール
エ	デバイスドライバ	バッファプール	スプーリング

Copyright 2005 BohYoh Shibata

参考文献

- 情報処理研究会『平成 10 年度 第 1 種情報処理技術者試験解答集』，電気書院，1997
- 『2001 年度版ソフトウェア開発技術者完全解答』，オーム社，2001
- 情報処理研究会『平成 9 年度春期版 第 2 種情報処理技術者試模範験解答集』，電気書院，1997
- 河村知信、金子崇『'99 年度版 第 1 種情報処理技術者合格完全対策』，経林書房，1999
- 河村知信『2000 年度版 第 1 種情報処理技術者第２種題＆分析』，経林書房，1999
- 藤本喜弘、河村知信『'98 年度春・秋試験対応版 第二種情報処理技術者試験第２種題＆分析』，経林書房，1997
- 財団法人 日本情報処理開発協会中央情報教育研究所 監修
『第二種共通テキスト②　ソフトウェア』，コンピュータ・エージ，1998

Copyright 2005 BohYoh Shibata