ソフトウェア開発技術者　受講生アドバイス

（直前でできること）

４月１８日（日）の試験本番まで，残りわずかとなりました。
今回は「直前でできること」と題して，午後試験対策として本番前に必ず準備・確認しておいていただきたい事項のアドバイス，ならびに午前試験に出題される重要キーワードのおさらい解説をお届けします。

試験まであと残りわずか･･･。受講生の皆さんは，午後（記述式試験）を中心に対策講座を通して着実に実力アップしたことでしょう。

今回は全体的に直前に実施できるポイント解説を記載しました。試験まで残り時間を計画的に活用し，不得意分野の克服に取り組んでください。

★午前のポイント（ここ３年間の出題が多い順に）
１．　コンピュータシステム、２．システム開発と運用、３．コンピュータ科学基礎
４．ネットワーク技術、５．データベース技術、６．セキュリティと標準化

★午後Ⅰのポイント
ソフトウェア工学、アルゴリズム、システム構成技術、システム開発、通信ネットワーク、データ構造・データベース、情報セキュリティ、システム評価

★午後Ⅱのポイント
アルゴリズム、システム開発

本試験では、これを問題事例に適用させ、その事例に特化した具体的な解を記述する必要があります。その糸口を得るために、いわゆる一般論・常識・定石としてこれらの要件を整理しておきましょう。

1　２週間前にできること

1.1 最後の仕上げに午前対策

最後の総仕上げとして、午前問題を一通り解いて知識をリフレッシュするのも効果的です。

IT 共通分野などは、とにかく試験当日に記憶にとどまっていればよいのです。直前におさらいして、午前問題をクリアしましょう。

＜巻末の重要キーワードは必ず暗記してください＞

1.2 最後の仕上げに午後対策

午後Ⅰの問題は平均２０分／問、午後Ⅱの問題は６０分の解答時間です。とにかく最後に一回は、その時間以上かかっても、納得いくまで解答を追い求めることが得点アップ、実力アップにつながります。

・知識の問題の場合：「知らない」と解けない問題は、解説を見て理解する事に重点を置く。

・その他の問題：問題を理解する事が重要。そのためには、「このようなデータの場合には、結果がこうなる。」というように、具体的なデータを作成し考えてみる事（計算問題でも、この値の場合にはこうなるであろう、という仮定をしておくこと）。

具体化（データを考える）、モデル化（Ａの場合、Ｂの場合・・・と様々なモデルを想定してみる）、

図式化（絵に書いてみる）を行うと文章の読み返しが減少し、理解もしやすく時間短縮になります。

1.3 本番に備えて最後の調整

本試験日に知力、体力、精神力のすべてがピークに達するように調整してください。友人や、家族の方の協力も使えるものはすべて使うべきです。前日は早めに床につきましょう。

試験会場に持参するものは前日までに用意して念入りにチェックしましょう。また、会場までの交通、駅からの地図を今一度確認してください。

2　試験当日

2.1 合格点狙いでよい

はっきり申し上げます。所詮はペーパー試験と割り切ってください。午前で７割、午後Ⅰで７割、午後Ⅱで７割、これで合格です。１００点を目指すのではなく、合格点狙いでゆきましょう。よく分からない小問の一つ、二つは早々と切り上げて部分点狙いでゆく位の大胆さが必要です。そういう問題に出くわしたならば、せめてキーワードの一つ吐き出して一矢報いる位の気構えで臨みましょう。空欄だけは避けましょう。解いているうちに答えが見つかることもありますから、気持ちの上で負けてはダメです。

制限時間内に合格ラインを超えるだけの解答をアウトプットすることを、試験当日の目標にしてください。徹頭徹尾、合格にこだわりましょう。

2.2 最後まで諦めない

試験に合格する人は「最後まで諦めないで全力を尽くす人」なのです。諦めた人から合格者は出ないのです。**「試験終了です」と言われて答案を試験官に渡す瞬間まで、答案を書く権利はあなたにある**のです。最後の１秒まで諦めないでください。

＜午後Ⅰ／Ⅱ試験テクニック＞

2.3 午後Ⅰ／Ⅱ対策

午後Ⅰ、午後Ⅱの問題の内容は以下の４つに分類されます。
午後Ⅱは１問でこれらの分類の設問が複数出題されると考えれば良いでしょう。つまり、午後Ⅰも午後Ⅱも基本的な対策には違いがないのです。ただ、午後Ⅱでは長くなる問題文を読みこなす力がより必要となりますので、
どの問題であっても後から見直す時間はないと考えて、必ず解いたその時に、再確認をすることが重要です。

１．知識の問題：知らない場合には解答出来ない問題。午前の知識を身につけると同時に、最新の技術に関しては、用語とその意味程度は日頃から確認しておくこと。
２．アルゴリズムの問題：フローチャート、もしくは擬似言語の形式で出題される問題。穴埋めであったり、データを設定して変数の推移を問うたり、ループ回数を求める設問となる。必ずテストデータを作成して、自分の解答が正しいかどうかを確認す

ること。

３．　設計に関する問題：システムの条件を問題で設定し、設計を行うもの。データベース設計で実施するＥ－Ｒ分析の問題が目立ちますが、これ以外にもクラスの設計やＤＦＤによる設計の問題が出題される可能性が高い。問題文には必ずヒント、あるいは解答そのものが記述されている場合が多い。よく文章を読むこと。

４．　計算問題：待ち行列、信頼性、通信速度等の問題が多く出題されている。計算方法をマスターし、勘違いや計算間違いをしないこと。得点源としたい。

（１）段落毎に本文を読み、要点をマークする。

次に、段落毎に本文を読みます。読み進めていくうちに、気になる箇所が幾つかあるはずです。

- 要件定義
- 数値や例を挙げてわざわざ具体的に書かれている事柄
- 業務の流れ（データのフローに変換）
- 例外的な事象
- 顧客や上司から指摘された問題点や改善提案、現行システムでは対応できないような事態
- 将来に問題発生が予見されるような不適切な処理、不自然な記述、極端な物言い

これら気になる箇所を、下線などを引いてマークしておきましょう。人によって、マークする箇所は異なる事があります。人それぞれ、経験やスキルが異なるからです。ある人にとっては「当たり前」であっても、異なる背景を持つ方にとっては「気になる箇所」と映るかもしれません。ですから、マークすべきポイントに唯一解はありません。「気になるな」と思ったらそこをマークすれば良いのです。とはいえ、慣れないと全部が全部気になってしまいマークした箇所がたくさん出てしまい、かえって要点がぼやけてしまう事もあります。ある程度、過去問題を解いて感触をつかむ必要があるでしょう。

このようにマークする利点は一体何でしょうか？午後Ⅱは多くの文章を読んで解析する必要があります。設問の答えに結びつく要点が、たくさんの情報量の中に埋もれてしまいがちです。しかも、制限時間が大変短い中で解答する必要がありますから、ひとたび取りこぼすと後でピックアップするのが難しくなります。

ですから、指定された時間内で効率的にポイントを洗い出すため、気になる箇所をマークしておき、目立たせておくのです。

本文を読んで行くうちに、ポイントがリストアップされている箇所に出くわす事があり

ます。このような箇所に丸数字や英文字などで番号を付けておきましょう。

また、図や表にも注目します。図表は、一目瞭然に情報を伝えます。また、詳細仕様、例外事項、制限事項などが密かに記述されている事が多いので、要チェックです。しかし、試験時間が限られていますから、最初に本文に目を通す段階では、図表の全ての文章や脚注に目を通す必要はありません。実際に設問を解く段階でもよいでしょう。

（２）設問全部に目を通し、戦略を立てる。

本文を読み終わったら、いきなり設問１の（１）から解くのではなく、設問全体に目を通しましょう。これから取り組むべき本当の相手は「設問」です。その真の姿を知り、戦略を立てて取り組む事により、高得点を狙う事ができるのです。今は頭の中に全体像が入っています。設問を読んで何かの手掛りやアイデアがひらめくかもしれません。それを余白に書きましょう。これから問題を１つ１つ解いてゆきますと、深みにはまってしまい、今せっかく思いついたことを忘れてしまうかもしれないからです。

それぞれの設問は独立性が高く、大抵１つか２つの見出しと関連があります。この段階でもリンク付け作業を行ってください。１つの設問の中で小問題は結合度が高いです。流れに沿って解くのもよいですし、一緒に考えてみたり、あえて順番を逆にしてみたりと、戦略を練ってみてください。

戦略を立てる事には、時間配分も含まれています。この設問はだいたいこれ位までに解いておく、という目標を書いておきましょう。一つの空欄の解を導くのに深みにはまってしまい、後の方の問題を解く余裕が無かった、という事が無いようにしましょう。どこに時間をかけ、どこを切り捨てても良いのか、その優先順位を判断する際に大切なのは、設問全体の戦略です。後の設問に結びつくようなポイントは是非とも時間をかけて解くようにしましょう。そのためには、設問を解く順番を変えてみてもよいです。

2.4 最後に

今回合格を目指すのか？次回で良いのか？
今、あきらめているのなら次回も合格は無理であると考える。理由は今以上に業務が忙しくなるから。
勉強すれば「合格する」可能性が非常に高い。勉強しなければ「合格しない」可能性が高い。
勉強すれば、合格の可能性を信じて本試験に挑戦する。

勉強して受験すれば自分の弱点が、再認識出来る。
　→合格すれば、次のステップに進めば良い。
　→不合格になっても、自分の弱点は認識出来るので、次回は弱点を補強すれば良い。

【合格するために】
１．合格する事を自分に誓う。
２．合格するために、何をどう勉強しなければならないか自分でもう一度考える。
３．この資料を参考にして、試験まで残された時間をどう過ごすかのスケジュールを作成する。
４．そのスケジュールに従い、勉強する。
５．毎日、スケジュールの確認をする。
６．遅れた場合にはスケジュールを変更し遅れを取り戻す方法を考える。
７．遅れの取り戻しが出来ない場合。→自分で出来る限りやるしかない。

【結果について】
・結果は自分の責任である。合格しようが不合格になろうが自分の結果である。
・受験する前からあきらめていては、絶対に合格しない。
・合格を信じて、勉強する事。
・合格すれば、次のステップに進もう！
・結果が不合格であっても、やるだけの事をすれば問題点が明確になる。
　→勉強したのに、合格出来なかった悔しさが残る。
　→（結果）次回は、より計画的に勉強し、弱点を考慮した学習となり、合格の可能性が高まる。
・途中で勉強しなくなると、どうせ合格は難しいと考えあきらめる。
　→あきらめたら、問題点として残るのは、勉強しなかった事だけになる。
　→次回どのように対応すれば良いか具体的な対策が出てこない。

皆さんの合格を祈っています。悔いの残らない勉強をしてください。

〈午前試験　重要キーワードのおさらい〉

過去の本試験午前問題にて繰返し出題されている，ＩＴ共通知識分野の重要キーワードのおさらい解説です。さっと斜め読みをしてみて，もし理解が曖昧なものがあれば，今のうちにテキスト等をあたって確認しておきましょう。

●コンピュータシステム

1　ハードウエア

１．１　情報素子

□メモリ素子の特徴（半導体）

ＳＲＡＭ：フリップフロップ回路で構成，高速，キャッシュメモリに使用

ＤＲＡＭ：リフレッシュが必要，遅い，主記憶に使用

ＳＤＲＡＭ：ＤＲＡＭの改良版，高速

□メモリ素子の特徴（集積回路）

ＣＭＯＳ：集積度を高められる

バイポーラ：動作速度がはやい

１．２　プロセッサアーキテクチャ

□高速化方式の特徴

パイプライン：異なる命令ステージの並行処理

スーパスカラ：同じ命令ステージでも並行処理可能

ＶＬＩＷ：長い機械語命令（関連する命令をあらかじめまとめたもの）による並列処理

１．３　メモリアーキテクチャ

□主記憶のアクセス時間短縮

メモリインタリーブ：主記憶を複数バンクに分割し並行アクセスすることによる高速化手法

キャッシュメモリ:（平均読み取り時間の計算は頻出）

平均読み取り時間 ＝キャッシュのアクセス時間×ヒット率 ＋主記憶のアクセス時間×（1－ヒット率）

１．４　補助記憶

□ＲＡＩＤの種類と特徴

ＲＡＩＤ０：ディスク２台へのストライピング／冗長性なし，狙いは性能向上

ＲＡＩＤ１：ディスク２台へのミラーリング（同一データ書込）

ＲＡＩＤ０＋１：性能と可用性の両立，但し高価

ＲＡＩＤ２，３，４：冗長符号用の専用ディスクを使用，この冗長化ディスクにアクセスが集中し性能上のボトルネックに／ディスクは最低３台必要

ＲＡＩＤ５：冗長符号（パリティ）用のディスクを特定せず，データディスクに分散格納することでボトルネックを解消／ディスクは最低３台必要

１．５　入出力アーキテクチャと装置

□パソコンの入出力インタフェース

ＳＣＳＩ：HDD，CD-ROM，スキャナ等の接続用

IDE（ATA)：HDD 接続用規格の古典

EIDE（ATAPI）：IDE の拡張版，HDD の容量制限緩和や CD-ROM サポート等が盛り込まれた

セントロニクス：もともとはプリンタ接続用だったが，双方向通信も OK

ＧＰＩＢ：計測機器接続用

ＵＳＢ：1.5/12Mbps の汎用シリアルインタフェース／キーボード，マウス，モデム等さまざまな周辺装置が接続可能

ＩＥＥＥ１３９４：100Mbps～の高速シリアルインタフェース

ＩｒＤＡ：赤外線通信，各種情報端末等のワイヤレス接続

１．６　コンピュータの種類とアーキテクチャの特徴

□スーパーコンピュータ

スーパーコンピュータはベクトルプロセッサやマイクロプロセッサを複数結合させて高速化をはかっている

2　基本ソフトウエア

２．１　オペレーティングシステム

□プロセス（タスク）とスレッド

プロセス（タスク）：ＣＰＵ資源，メモリ資源の割り当て単位／実行可能状態，実行状態，待ち状態の三つの状態間を遷移

スレッド：プロセス（タスク）内に複数存在／メモリ資源（アドレス空間）は共有，ＣＰＵ資源（スタック，プログラムカウンタ，レジスタ）は別々に割り当てられ並行処理

□割込み

内部割込み／外部割込みの区別：プログラムの内側（＝プログラムによって発生する割込み）か，外側（＝ハードウエアによって発生する割込み）かの違いによる分類

内部割込み：ＳＶＣ割込み（システムコール），プログラム割込み

外部割込み：入出力割込み，タイマ割込み，マシンチェック割込み，コンソール割込み

□仮想記憶

仮想記憶の分類：

セグメンテーション（方式）：アドレス空間の分割を可変長の「セグメント」単位に行う方式

ページング（方式）：アドレス空間の分割を固定長の「ページ」単位に行う方式／主記憶装置から補助記憶装置への主記憶内容の退避（ページアウト）や呼び戻し（ページイン）

関連用語：

スラッシング：実記憶容量が充分でない場合，プログラム多重度の高まりにより主記憶装置～補助記憶装置間のページ入替の頻発

スワッピング：主記憶装置から補助記憶装置への主記憶内容の退避（スワップアウト）や呼び戻し（スワップイン）

ページフォールト：ページング方式で，必要なページが主記憶に存在しない場合に発生する割り込み

３　システム

３．１　システムの構成技術

□信頼性向上のためのシステム構成

オンラインシステムシステム構成：

デュアルシステム：システムを完全二重化，結果の照合／原子力プラントのオンラインシステム等

デュプレックスシステム：故障時に主系（現用系）から副系（待機系，予備系）に切り替え／平時は主系でオンライン処理，副系でバッチ処理

現用系－予備系への切り替えの仕方：

ホットスタンバイ：現用系と同じシステムを予備系でも起動して待機

ウオームスタンバイ：予備系ではＯＳだけ起動して待機

コールドスタンバイ：予備系の電源を切って待機

３．２　システムの性能

□Ｍ／Ｍ／１の待ち行列モデル（頻出）

待ち行列〔概要〕

（分野：コンピュータシステム→システムの構成と方式）

待ち行列理論では，その構成要素を次の３つに分けて考えます。

・集団：やがて窓口にやってくる，サービスを受けようとする集団

・待ち行列：窓口が処理中でふさがったときにできる並び（窓口が空き次第，順番にサービスを受ける）

・窓口：実際にサービスを提供するところ

待ち行列の基本的な考え方は，「母集団自体はき

わめて大きいが，ある時点に着目してとらえると，実際にサービスを受けに来ている人はそのごく一部である」という事実に基づいて，「経済的・統計的な観点から窓口の数を決めよう」というものです。

母集団内の個々の要素を「トランザクション」と呼びます。銀行などの窓口の場合は待っているお客一人一人，コンピュータ処理の場合はＣＰＵやＩ／Ｏなどのリクエストがこれに該当します（ついでに言えば，コンピュータ処理の場合，待ち行列は典型的には「入出力キュー」が該当します)。

待ち行列の長短は，サービスを受けに来る人の到着頻度とか，窓口でのサービスのさばき具合，あるいは窓口の数等によって左右されます。これらについて，ＡＥ・ＰＭ・ＳＡ午前共通問題では，次の前提条件のもと出題されます。

・一定時間内の到着トランザクション数は全くランダム（これをポアソン到着と呼ぶ）
・あるトランザクションの処理時間の統計をとると指数分布のカーブを描く（この場合，一定時間内に処理できるトランザクション数に着目すると全くランダム）
・サービス窓口の数は一つ

このような，待ち行列を左右する条件を表記するための記法が「ケンドールの記法」です。それを用いて上記の前提条件を表したものが，問題文中に示される「Ｍ／Ｍ／１型待ち行列」という表現です。

〔待ち行列に関する公式〕

参考書にはいろいろな公式が掲載されていますが，以下の公式とそれぞれの意味を理解していれば充分です。

(1) 平均到着率：$\overset{ラムダ}{\lambda}$，平均到着間隔：Ta

ある一定時間当たりに到着するトランザクション数が，平均到着率（λ）です。「単位時間あたりのリクエスト数」とか「単位時間あたりの平均到着数」などと表現するテキストもあります。

同じ事柄を違った観点で表現したものが，平均到着間隔（Ta）です。あるトランザクションが到着してから，次のトランザクションが到着するまでの時間間隔の平均です。

λと Ta には次の関係があります。

$$\lambda = 1/Ta$$

(2) 平均サービス率：$\overset{ミュー}{\mu}$，平均サービス時間（平均処理時間）：Ts

ある一定時間当たりにサービス可能なトランザクション数が，平均サービス率（λ）です。「単位時間あたりに処理可能なリクエスト数」とか「単位時間あたりの平均処理数」などと表現するテキストもあります。

同じ事柄を違った観点で表現したものが，平均サービス時間（Ts）です。あるトランザクションが窓口に入って処理開始してからそのトランザクションが処理を終了し窓口を出る（それと同時に，次のトランザクションが窓口に入って処理開始する）までの時間間隔，すなわち「トランザクション１件あたりの処理時間」，その平均です。

μと Ts には次の関係があります。

$$\mu = 1/Ts$$

(3) 窓口利用率（あるいは単に利用率）：$\bar{\rho}$

ある一定時間当たりに窓口が利用されている確率が，窓口利用率（ρ）です。次の式で表せます。

$$\rho = \lambda / \mu = Ts/Ta$$

「窓口のキャパシティを越えた数のトランザクションが到着」，つまり「一定時間あたりに到着するトランザクション数（λ）が，一定時間あたりにサービス可能なトランザクション数（μ）を越えた状態（ρ＞1)」になると，待ち行列が際限なく大きくなることがわかります。

(4) 処理中も含めた待ち行列中の平均トランザクション数：L

次の式で定義されます。

$$L = \rho /(1- \rho)$$

なお，これに類似のものとして，「処理中を含めない待ち行列中の平均トランザクション数（＝純然たる待ち行列の長さ)：Lq」なる概念もありますが，こちらは試験対策としては不要と思われます。

(5) 平均待ち時間

到着してからサービスを受けるまでの待ち時間，言い換えれば順番待ちの時間です。自分の前にいるトランザクション数（窓口内にあるものも含む）に平均処理時間を乗ずれば求められますね。

$$Wq = L \times Ts$$

(6) 平均応答時間

到着してから処理が完了するまでの総待ち時間です。順番待ちの時間に自身の処理時間を加味すれば求められますね。

$$W = Wq + Ts$$

マスターすべき公式はこれだけです。確認の意味では具体的問題にチャレンジしてみる必要があるでしょう

３．３　システムの信頼性・経済性

□システム信頼性設計の考え方

フェールセーフ：障害が発生した場合，その影響が安全サイドに働くように設計すること（例：交通管制システムの障害時，混乱を回避するためにすべての交通を停止する指示が出るよう設計）

フェールソフト：障害が発生しても，それによる処理能力の低下はあるものの，機能停止せずに維持できるように設計すること（例：デュアルシステム）

フォールバック（縮退)：フェールソフトシステムで，障害が発生した装置を切り離して運転している状態

フールプルーフ：意図しない使われ方をしても故障しないこと

４　システム応用

４．１　データベースの応用

□データウエアハウス

データウエアハウス：膨大な生データの集まり／ここからＯＬＡＰツールやデータマイニングツール等で意思決定用の情報を抽出

ＯＬＡＰ：対話型データ分析

データマイニング：大量データから規則性・法則性を見つける

データマート：多角的データ分析をサポートするための目的別データベース

●システム開発と運用

１　システムの開発

１．１　言語

□プログラムの特徴

リエントラント：再入可能／データ域を別々に用意することで，複数プログラムからの同時呼び出し可能

（シリアリ）リユーザブル：（逐次）再使用可能／自身でデータ域の初期設定をすることで，再ロードせずに何度も使用可能

リロケータブル：再配置可能／主記憶の任意のアドレスに配置して実行可能

リカーシブ：再帰／自分の中から自分を呼び出し可能

１．２　開発手法

□ソフトウエア開発プロセスモデル

ウオータフォールモデル：開発を上流から下流への一方向に進める

スパイラルモデル：開発プロセスを繰返し，各繰返しにおいてコストや品質などの評価を行い，リスクを最小にするプロセスをとる

インクリメンタル（成長型）プロセスモデル：ウオータフォールモデルを繰返し，機能を段階的に提供

プロトタイピングモデル：開発初期段階での試作品による確認により，後工程での手戻りを防ぐ

ＲＡＤ：“早く，安く，高品質”のシステム開発を目的とした短期システム開発手法

ラウンドトリップ：オブジェクト指向開発において，分析・設計～プログラミングを行き来しながらトライアンドエラーで完成させていく方法

□プロセス成熟度モデル（CMM）

ソフトウエア生産部門におけるプロセス（作業）水準

レベル０：プロセスが未定義

レベル１：初期･･･勘に頼っている

レベル２：反復可能・・・統計値等が活かされている

レベル３：定義･･･プロセスが統合的な組織運営によって実施されている

レベル４：管理･･･プロセス目標を念頭においた制御・改善の仕組みがある

レベル５：最適化：ビジネスニーズを念頭においた改善の仕組みがある

１．３　要求分析・設計手法

□データ中心アプローチ

プロセス（機能）中心設計：データの流れ・機能に着目した，従来型の設計

データ中心設計：データ構造をベースにプロセス設計を行う手法

オブジェクト指向設計：データとプロセスを一体化（カプセル化）／データ中心設計の発展形

１．４　プログラミングの手法

□オブジェクト指向の概念

インヘリタンス（継承）：サブクラス（特化）～ス

ーパクラス（汎化）の関係〔例：乗用車，バス，トラック→自動車〕／サブクラスはスーパクラスの属性を引き継ぐ
集約：構成部品と製品の関係〔例：駆動装置，車体，車輪→自動車〕
ポリモルフィズム（多様性）：同一メッセージを送ってもオブジェクトごとに動作が異なること
カプセル化：データと操作（メソッド）をオブジェクトとして一体化

１．５　テスト・レビューの手法

□レビュー技法

インスペクション：組織的・制度的なレビュー，モデレータ（進行役）中心に進行
ウオークスルー：自主的レビュー，作成者中心
ラウンドロビンレビュー：レビュー参加者が持ち回りでレビューのリーダを担当する方法

１．６　開発管理

□システム開発工数見積り

ファンクションポイント法：機能数（ビジネスアプリケーション開発では画面数や帳票数などがその目安になる）から開発規模を算出
ＣＯＣＯＭＯ：過去の生産性データを参考に開発規模（行数）から開発人月を推定

２　システムの運用と保守

２．１　システムの運用

□非常用電源装置

UPS：Uninterruptible Power Supply；電源の瞬断や短時間の停電用
自家発電装置：長時間の停電用／CVCF と組み合わせて使用
CVCF：Constant Voltage Constant Frequency；自家発電装置の出力（電圧・周波数の変動がある）を安定させるために使用
AVR：Automatic Voltage Regulator；自動定電圧装置／電圧変動が小さい場合に有効

●セキュリティ

１　セキュリティ対策

１．１　機密保護・改ざん防止対策

□公開かぎ暗号方式

（分野：セキュリティと標準化→セキュリティ）

公開かぎ暗号方式以前から広く用いられていたのが，送信側と受信側のみが知っている，かつ両者だけの秘密のかぎを用いる「秘密かぎ暗号方式」です。この「秘密かぎ暗号方式」は，かつて「慣用暗号方式」と呼ばれていましたし，さらに送信側と受信側で共通のかぎを用いるため「共通かぎ暗号方式」と呼ばれることもあります。

さて，情報の秘匿のために暗号化する，という機能そのものに即して考えれば，秘密かぎ暗号方式でも問題はありません。しかし，情報通信の世界で暗号を使用するにあたり，本文の暗号化を行う前の問題として，「秘密であるべきかぎを，いかにして秘密裡に相手に送り届けるか」という矛盾が生じました。この答えが，公開かぎ暗号方式です。

公開かぎ暗号方式は，非可逆関数を用いています。私達は中学・高校などで「逆関数」を学びました。たとえば，$y=2x-4$ に対する $y=x/2+2$ とか，あるいは $y=5x^2$ に対する $y=\mathrm{sqrt}(x/5)$ の類です。逆関数は元の関数よりも複雑という印象がありますが，公開かぎ暗号方式は，この逆関数が複雑どころかそれを見出すことが（事実上）不可能，という特殊な数式（と数学の体系・理論）を用いて実現されています。

この特殊な数式に関する２つ（一対）のパラメータを，鍵として用意します。この一対の鍵それ自体は，元々はどちらがどちら（どちらが暗号用でどちらが復号用）といった違いはありませんが，面白いことに一方のかぎを用いて暗号化した場合，他方のかぎを用いないと復号できない（暗号化した際に用いたかぎでは復号できない），という性質があります。

この一対のかぎのうち一方を「秘密裡に相手に送り届けることができない」情報通信などを用いて相手に送り届けます。この過程で第三者が盗聴する危険性も当然ありますが全くおかまいなしです。この送り届けたほうのかぎを「公開かぎ」と呼びます。そしてもう一方の手元の鍵は，自分だけが知るものです。これを「秘密かぎ」と呼びます。

ここでポイントとなるのは，次の２点です。

- 情報の秘匿を目的として公開かぎ暗号方式を用いる場合，一対の鍵を用意するのは，受信者の側である
- 秘匿すべき情報のやりとりに先立って，あらかじめ受信者から送信者に対し，鍵（受信者の公開鍵）の受渡しを行う

つまり，「自分に秘密裡に情報を送り届けて欲しい」と思う側（受信者）が一対の鍵を用意し，そのうちの片方の鍵をあらかじめ相手（送信者）に送り届けておくわけです。そして，受信者から送信者に対するかぎのやりとりが済ませてしまった後，実際の秘密情報のやりとりを行うわけですが，情報の送信者は「受信者の公開かぎ」を用いて情報を暗号化して送信，その受信者は，「受信者（自身）の秘密かぎ」を用いて暗号化された情報を復号します。

さて，ここで悪意の第三者に「受信者の公開かぎ」と「暗号化して送信した情報」の両方を入手されたとします。しかし送信者は第三者が入手したのと同じ「受信者の公開かぎ」によって暗号化していますから，悪意の第三者は情報を復号することができないのです！

ところで，一方の鍵を公開しても危険がないのは，前述した不可逆関数の特質によります。たとえばわれわれは，元の関数（たとえば「二項を乗ずる」という計算式）と，二項のうちの一項（たとえば 5），ならびにその答え（たとえば 10）がわかれば，二項のうち残りの一項がいくつ（この場合 2）であるかがわかります。それは，「二項を乗ずる」の「逆関数」である「積を一項で割る」によって求めることになります。ところが，公開かぎ暗号方式の世界では，この逆関数を見出すことができません。それゆえ，たとえ元の関数，関数で用いる一項，そしてその答えのうち，その一部を公開した（あるいはたとえそのすべてが既知になった）としても，秘密である残りの一項は決して求められないのです。それは逆関数を見出すことができないからです。これが公開かぎ暗号方式のからくり（のイメージ）です。

このような「かぎ配送にかかる問題点」をクリアした公開かぎ暗号方式の欠点は，計算の複雑さ（＝計算量の多さ，すなわち遅さ）です。それゆえ，現実の情報通信では，そのすべてを公開かぎ暗号方式で行うのではなく，秘密かぎ暗号方式（＝公開かぎ暗号方式より高速）と適材適所で使い分けしています。その具体例が「ＳＳＬ」や「ＰＧＰ」などのセキュリティプロトコルです。

ここまで理解できたなら，できれば上述のセキュリティプロトコル，そして電子署名や電子認証

の概要について確認しておきましょう。そこまでやっておけば，暗号化の仕組みについては完璧です。

□公開かぎ暗号方式の応用

(1)機密保護

〔送信者〕本文を受信者の公開かぎで暗号化

〔受信者〕受信者の秘密かぎで復号

(2)送信者の認証（電子署名）:

〔送信者〕署名を送信者の秘密かぎで暗号化

〔受信者〕送信者の公開かぎで復号（復号できれば送信者は秘密かぎの持ち主つまり本人であることがわかる）

(3)送信者の認証＋改ざん有無の確認

〔送信者〕本文からハッシュ関数を使ってメッセージダイジェストを作成し，これを送信者の秘密かぎで暗号化／本文は別途何らかの方法で暗号化する等して送信

〔受信者〕何らかの方法で暗号化する等して送信された本文からハッシュ関数を使ってメッセージダイジェストを作成／これと送信者の公開かぎで復号することで取り出したメッセージダイジェストと比較（一致していれば改ざんがないことがわかる）

２　リスク管理

２．１　リスク分析

□リスク管理

リスク管理の基本的考え方：一定のコストでリスク発生の場合の損害を最小にするためのプロセス

リスク管理の方法：移転（契約等を通して他者に転嫁），保有（準備金積み立てや安全度を見込むことで受容），分離等

３　ガイドラインと関連法規

３．１　セキュリティに関するガイドライン

□コンピュータウイルスが持つ機能

コンピュータウイルス対策基準による定義：

自己伝染機能（他のプログラムやシステムに自身をコピーする）

潜伏機能（発病まで症状を出さない）

発病機能（破壊や利用者の意図しない動作を行う）

●標準化

１　開発と取引の標準化

１．１　開発プロセス，取引プロセスの標準化

□ＩＳＯ９０００シリーズ

ＩＳＯ９００１～９００３：品質保証モデル／この中でソフトウエア開発・供給・保守に適用されるものはＩＳＯ９００１

ＩＳＯ９０００－３：ＩＳＯ９００１をソフトウエア開発・供給・保守に適用するための指針

ＩＳＯ９００４：ＩＳＯ９００１を取得した企業が継続的改善を目指す際の指針（ＩＳＯ９００１等と対をなす関係）

２　情報システム基盤の標準化

２．１　ソフトウエアの標準化

□ＣＯＲＢＡ

ＣＯＲＢＡ：機能単位に分割したアプリケーション（オブジェクト）を分散システムで実行・通信・管理する技術仕様

ＯＲＢ：ＣＯＲＢＡのソフトウエア及び機能

ＯＭＧ：ＣＯＲＢＡを制定した標準化団体

３　データの標準化

３．１　ファイル形式の標準化

□文書の構造化記述言語

ＳＧＭＬ：さまざまな文書構造の記述について標準化した国際規格／機能が豊富すぎて実装が困難

ＸＭＬ：いわば“実装可能なＳＧＭＬ”／Ｗｅｂ対応と拡張性を考慮した，ＳＧＭＬのサブセット

ＨＴＭＬ：これもＳＧＭＬのサブセット／Ｗｅｂの記述用として広く普及（この文書もＨＴＭＬで記述されている）

４　標準化組織

４．１　標準化組織

□標準化団体

ＡＮＳＩ：米国規格協会

ＩＳＯ：国際標準化機構／工業・技術に関する国際規格の統一と国家間調整

ＩＴＵ－Ｔ：国際電信電話諮問委員会／電気通信の標準化に関する勧告を行う国連機関

ＩＥＥＥ：米国電気電子工学会／ＬＡＮ等にインタフェース規格制定に尽力

※標準化は「キーワードのオンパレード」です。前述の項目だけでなく，ＩＴ共通知識体系を参照し，漏れのないように確認しておきましょう。

●ネットワーク技術

※最低限覚える必要がある用語ですので、必ず言葉の内容だけではなく関連性を理解してください。

□アプリケーション層のプロトコル

1.　Ternet

TCP/IP プロトコルのネットワークで、遠隔地から他コンピュータに接続するためのプロトコル。端末間でオクネット単位の通信を実行することでリモートの仮想端末機能を実現する

２．ＦＴＰ

ファイル転送プロトコル。TCP/IP において、クライアントとサーバーの間でファイルを転送するためのプロトコルである。

３．SMTP

インターネットなどの TCP/IP ネットワークで、電子メールを送信するためのプロトコルである。

４．POP３

インターネットで電子メールを受信するために量されているプロトコルである。SMTP で送信された電子メールはメールサーバーのメールボックスに保存されているが、これを受信者が取り出すために POP3 が多く使われる。

５．MINE

インターネットの電子メールで半角英数字の ANCII コード以外の漢字コードやマルチメディアデータを送るための拡張機能

６．DHCP

クライアント側の TCP/IP 環境設定（IP アドレスの設定他）を自動的に実施する

７．HTTP

www でサーバーとクライアントの web ブラウザの間で、HTML で書かれたハイパーテキストを送受信するための TCP/IP の上位のプロトコル

□トランスポート層、ネットワーク層のプロトコル

１．TCP

コネクション型の通路路を提供するプロトコル

２．UDP

コネクションレス型の通信を行うプロトコル

３．IP

インターネットプロトコルの略

□データリンク層のプロトコル

１．PPP

２地点間でポイントツーポイント接続を行うデータリンク層のプロトコル

２．ARP

LAN 環境において IP アドレスからハードウェアがもつ MAC アドレスを調べるときに利用する

□伝送速度

データ伝送速度は１秒間に伝送できるビット数を示し、単位はビット／秒（b/s,bps）である。データ転送速度ともいい、例えばWANやLANの通信速度を表す場合はｋビット/秒やＭビット/秒、Gビット/秒などをもちいる。

伝送速度×時間＝伝送データ量（伝送データ量÷伝送速度＝時間）となる

□誤り検出

データ伝送などで誤りが発生したことを検出すること

① パリティチェックで１ビット誤りを検出する
② ECCで複数ビットの誤りを検出する
③ デュアルシステムのように２つのシステムのように２つのシステムの処理結果を比較して、誤りを検出する

□LANのアクセス制御方式

１．CSMA/CD方式

ツリー型やバス型のLANに適用される媒体アクセス方式である

２．トークンパッシング方式

伝送路上を送信許可をあたえるトークン（識別信号）が巡回しており、そのトークンを獲得した局が送信権を得ることができる方式である

３．時分割多重化装置を用いて、ＬＡＮで接続された各装置に固定的に小さな時間（タイムスロット）を割り当てる方式

□インターネット関連

１．WWW

ハイパーテキスト形式のマルチメディア情報検索システム

２．HTML

インターネットのWWWのページを書くためのマークアップ言語。マークアップ言語とは、テキスト中にマークをつけてテキストの情報を埋め込んだり、テキストに対する操作を指定するものである。

３．URL

インターネットのホームページファイルの場所を表したもの。ｗｅｂブラウザでＵＲＬを指定する

●データベース技術

※最低限覚える必要がある用語ですので、言葉の内容だけではなく関連性を理解してください。

□関係データベース

１．選択

表から条件を満たす組だけを抽出して、新しい表を作成する操作

２．射影

表から必要なフィールドだけを取り出して、新しい表を作成する操作

３．結合

特定のフィールドの値をもとに、複数の表のレコードを結びつけて、新しい表を作成する操作

□データの正規化

※受注伝票や社員名簿などを例にとって具体的に実施してみてください

関係データベースにおいて、表から集合や配列などを取り除き、データの整合性を保てるような表にすることを正規化するという。

１．非正規形

表や伝票など、ひとつのコードにおいて

２．第１正規化

繰り返し項目をそれぞれ別の行に分離する

３．第２正規化，第３正規化

別表に持つべきものの分離。

第２正規化は、主キーの一部だけに依存する列の

分離する。第３正規化は、主キー以外の列に依存する列の分離する

□ＳＱＬ（代表的な命令）

関係データベースは行と列からなる２次元の表であるが以下の命令で操作を行うことができる

１．ＳＥＬＥＣＴ　データの照会

２．ＩＮＳＥＲＴ　挿入

３．ＤＥＬＥＴＥ　削除

４．ＵＰＤＡＴＥ　更新

※二次元の表からレコードを抽出するには、次の命令を用いる

ＳＥＬＥＣＴ　項目名　ＦＲＯＭ　表名　ＷＨＥＥ　条件

項目名：抽出したい項目の名前（２個以上も可）

表名：抽出のもととなる表の名前（２個以上も可）

条件：抽出すべきレコードの選択条件（ＡＮＤやＯＲを用いてよい）

●コンピュータ科学基礎

□数値変換とデータ表現

※言葉の意味を理解して実際に数値を当てはめて表現してみてください。

１．固定小数点

少数点の位置が暗示的に固定されている数値、

数値の表現。

２．浮動小数点

仮数に基数を指数でべき乗した値をかけて、実数を表した数値。または数値表現。

□論理演算

※言葉の意味を理解して実際に数値を当てはめて表現してみてください。

①　どちらか一方でも１なら１になる論理和（ＯＲ演算)、

②　両方が１のとき１のなる論理積（ＡＮＤ演算）

③　１のとき０，０のとき１になる否定（ＮＯＴ演算）が基本となる

④　異なるとき１になる排他的論理和演算（ＥＯＲ演算、ＸＯＲ演算）

⑤　一致するとき１になる一致演算

⑥　どちらか一方でも１なら０になる否定論理和演算（ＮＯＲ演算）

⑦　両方が１のとき０になる否定論理積演算（ＮＡＮＤ演算）

□形式言語

※言葉の意味を理解して実際に数値を当てはめて表現してみてください。

１．逆ポーランド記法

数学上の式を表すとき、演算子がオペランドの後ろに置かれる表記法

２．ＢＮＦ記法

①　対象となる言語の記号を終端記号

②　ＢＮＦで使用する記号を非終端記号と呼び、＜＞で囲んで表す

□データ構造

データとデータの論理的なつながりを表したもの

１．配列（線形リスト）

データが順序よく並んだもの

２．キュー（先入れ先出しリスト）

先にいれたデータから先に取り出されれるもの

４．連結リスト

・単方向リスト

次のデータのポイントだけをもつもの

・双方向リスト

前と次のデータへのポイントをもつもの

・環状リスト

データが環状連結されているもの

□木構造

※言葉の意味をよく理解して実際に表現してみてください。

①節から出る枝が２本以下の木を２分木

②節から出る枝が３本以上の木を多分木という。

節の挿入や削除を繰り返しても、節の階層の差を１以内にして、左右の節の数がほぼ等しくなるようにした木を平衡木（バランス木）という

１．２分木

・２分木探索木

節の左右で大小関係を割り振った木

・部分順序木

階層で大小関係を割り振った木

・完全２分木

節の左右の枝の数が等しい木（平衡木）

・ＡＶＬ木

節の階層の差が１以内の木（平衡木）

２．多分木

・Ｂ木

葉の深さが同じ木

□整列アルゴリズム

※言葉の意味を理解して実際に数値を当てはめて表現してみてください。

１．選択ソート

配列の中から、最小の値を選んで並べていくという整列方法である。最小値ソートとも呼ばれる。手順としては、配列の最左端の要素を基準に、右に並ぶ各要素と比較しながら最小値を見つけ出し、最左端に格納する。これによって、配列の最小値が最左端に確定する。以上の処理を配列の最右端から一つ前までの要素に対して繰り返す。

２．バブルソート

配列の隣同士を比較しながら、並べ換えるという整列方法である。このため隣接交換法と呼ばれる場合もある。手順としては、配列の最左端の要素とその隣の要素を比較しながら、配列の最右端になるまで繰り返す。これによって、配列の最右端に最大値が確定する。以上の処理を最左端の次の要素まで繰り返す。

３．挿入ソート

配列の要素の最左端からひとつづつ取り出し、該当する箇所に順番に挿入していくというアルゴリズムである。手順としては、配列の最左端の要素とその隣の要素を比較し、それらの大小によって位置を仮に確定する。次にそのもうひとつ右の要素を取り出し比較することによって、三つの要素の順番を仮に確定する。このとき、必要ならば間に挿入することもある。以上の処理を最右端になるまで繰り返す。

４．クイックソート

ある基準となる要素に対して、それより小さい要素を左側に、それより大きい要素を右側に集めることを繰り返すアルゴリズムである。手順としては、まず配列の真中を基準値にして、それより左側に小さい要素、右側に大きいい要素を集める。次にそれらの部分配列に対して再び基準値を設定して振り分ける。以上の処理をすべての配列の要素を網羅するまで繰り返す。

□探索アルゴリズム

※言葉の意味を理解して実際に数値を当てはめて表現してみてください。

１．線形探索

逐次探索、順次探索とも呼ばれ、物理的な順番のもとに検索を行うというアルゴリズムである。手順としては、虱潰しによる方法があるが、これには、配列すべての要素をくまなく探すという意味が含まれている。

２．２分探索

配列に格納されている要素がある順番に並んでいる場合に適用でき、配列を二分しながら探索を進

めるアルゴリズムである。手順としては、配列のうち分割する位置（基準値）を計算する。これを基準に探査値が左側か右側のどちらにあるかを判断する。以上を繰り返しながら、配列の分割範囲を狭めていく。その繰り返しの中で検査値と一致するか、一致しないまま隣接した要素同士の比較を行った時点で終了する。

試験当日まであとひと頑張り，最後の追い込みを。そして試験当日は途中退場せず，試験時間満了まで全力を尽くしましょう。

あなたなら必ず合格できます！　合格を心から成功を祈っています！！