平成１８年度　春期　テクニカルエンジニア（情報セキュリティ）　午後Ⅱ問題

問１　システム統合時におけるアクセス制御の設計に関する次の記述を読んで，設問 1～3 に答えよ。

X 社は，従業員数 500 名ほどの中堅エレクトロニクス関連総合商社である。X 社では，これまでエレクトロニクス製品を扱う EL 事業部と，コンピュータハードウェア関連製品を扱う IT 事業部とで，事業を展開してきた。今後は，事業の強化・拡大と社内体制強化という二つの方針に従って，より一層の躍進を目指すことにした。

事業の強化・拡大の具体策として，ソフトウェア開発・販売会社の Z 社を買収し，SW 事業部とした。社内体制強化の施策の一つは，社内における人材の有効活用である。このために，各事業部の人事担当者を集約して本社管理部人事グループとし，事業部の枠にとらわれずに全社的見地から人材活用を図ることにした。併せて，人事担当者と同様に各事業部に所属していた経理担当者，総務担当者を集約してそれぞれ経理グループ，総務グループとし，人事グループとともに統合して本社管理部とした。X 社は，EL，IT，SW の 3 事業部と，本社管理部から構成されている。

社内体制強化のもう一つの施策は，重複している情報資産の排除及び集約化である。事業部ごとに構築してきた情報システムを見直して重複をなくすとともに，情報を集約して有効に共有するため，システムの統合を行うことになった。また，各事業部で管理してきた顧客情報も，全社的なビジネス戦略推進の観点から，本社管理部で集中管理することになった。

システムの統合において考慮すべき点は，昨今の情報漏えい事件などを踏まえ，機密情報の管理をこれまで以上に徹底することである。そこで，システム統合における最も重要な要件として高度なアクセス制御機能が必須と判断された。そしてシステム統合を担当する開発チームが組織され，検討が始まった。

〔情報セキュリティ基本方針及び対策基準とシステム要件〕

SW 事業部の設置をきっかけに，X 社ではセキュリティ強化のため，情報セキュリティ基本方針及び対策基準の見直しを行った。図 1 に示すように，改定後の X 社情報セキュリティ基本方針及び対策基準では，情報資産の情報区分をより細分化することによって情報の機密管理を強化している。

基本方針

（省略）

〔情報資産の取扱い〕

1. 情報資産は，資産価値や機密度に応じて適切な情報区分に格付けする。
2. 情報資産に対する情報区分の格付け責任者は，情報資産作成者の部門長とする。

（省略）

〔情報資産に対するアクセス制御〕

1. 情報資産は，情報区分及び利用者のアクセス区分に応じて適切なアクセス制御を実施する。
2. 情報資産に対する情報区分の格付けや利用者のアクセス区分の付与（変更を含む）と，情報区分と利用者のアクセス区分のシステム設定作業は，別の管理者が実施する。

（省略）

対策基準

〔情報資産の情報区分〕

1. 情報資産の情報区分は，取扱レベルとカテゴリから構成する。
2. 情報資産の取扱レベルは，機密度の高い順に次の 6 段階に分類する。
   - 最高機密，極秘，秘密，社外秘，取扱注，一般
3. “社外秘”以上の取扱レベルの情報資産は，機密情報資産とする。
4. カテゴリは，情報資産が帰属する事業部，部，プロジェクトなどの業務遂行単位を識別するために付与する情報資産の属性である。

〔利用者のアクセス区分〕

1. 利用者のアクセス区分は，権限クラスと一つ以上のカテゴリから構成する。利用者には同時に一つ以上のアクセス区分を付与できる。
2. 利用者の権限クラスは，情報資産へのアクセス権限の高い順に次の 6 段階に分類し，職務権限に応じて付与する。
   - クラス 6，クラス 5，クラス 4，クラス 3，クラス 2，クラス 1
3. 利用者のアクセス区分中の各カテゴリは，利用者が情報資産へのアクセス許可を得るために必要となる情報資産の情報区分のカテゴリを示すために付与する利用者の属性である。

〔情報資産に対するアクセス制御〕

1. 各機密情報資産の取扱レベルと利用者の権限クラスは，“最高機密”と“クラス 6”，“極秘”と“クラス 5”，“秘密”と“クラス 4”，“社外秘”と“クラス 3”に，それぞれ対応させる。
2. 機密情報資産へのアクセス可否は，情報資産の情報区分と利用者のアクセス区分によって判断する。具体的には次のアとイ，又は，アとウの基準に基づいてアクセスを許可する。
   ア．利用者のアクセス区分中のカテゴリと，アクセス対象の情報資産の情報区分中のカテゴリとが一致する。
   イ．情報資産にアクセスする利用者の権限クラスが，それに対応する情報資産の取扱レベルに等しいか高い場合だけ，参照アクセスを許可する。
   ウ．情報資産にアクセスする利用者の権限クラスが，それに対応する情報資産の取扱レベルに等しい場合だけ，書込みアクセスを許可する。
3. 機密情報資産以外の情報資産へのアクセス可否は，利用者の必要度に応じて情報資産の所有者が与えたアクセス権によって判断する。

（以下，省略）

図１　改定後のＸ社情報セキュリティ基本方針及び対策基準

システム統合の対象である，EL 事業部，IT 事業部及び SW 事業部の人事管理システム，顧客管理システム及び文書管理システム（以下，これらを業務システムという）は，この情報セキュリティ基本方針及び対策基準に準拠して，再構築及び運用されることになった。

IT 事業部の人事管理システム及び顧客管理システムのアプリケーションプログラムは，Y 社に委託して開発されたもので，人事情報及び顧客情報は関係データベース（RDB）に格納されている。

EL 事業部及び SW 事業部の業務システムは廃止し，EL 事業部及び SW 事業部の人事情報，顧客情報，ソフトウェアの設計資料などの文書は IT 事業部の業務システムに移行する。移行後の IT 事業部の業務システム（以下，全社業務システムという）は，図 2 に示すように全社で利用する。

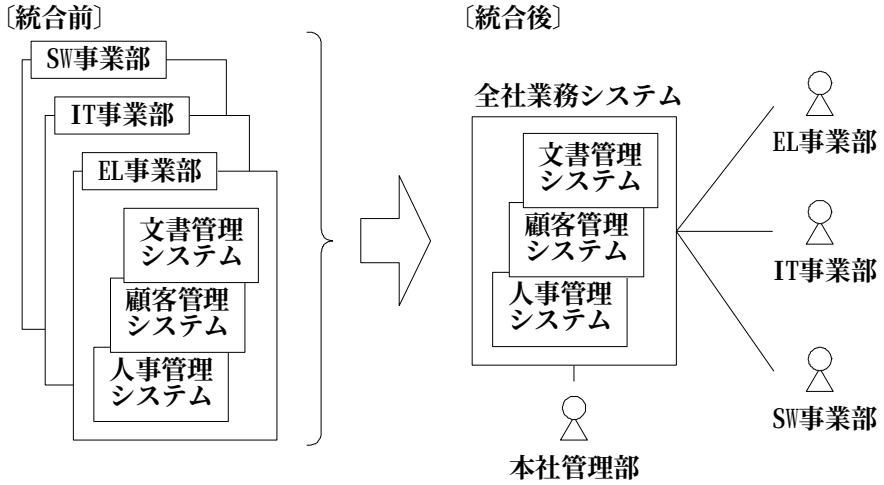

図２　システム統合のイメージ

全社の人事情報を本社管理部で一括して管理することになったが，人事権及び人事管理の責任は各事業部に帰属するという方針は，従来と同様である。人事管理システムは，この方針を維持しつつ，システム統合後も本社管理部の業務が円滑に行われるようなアクセス制御が必要となる。

文書管理システムは，従業員が PC で作成した，新製品などの企画文書，設計書，図面ファイルなどの文書を保管する。これらは，システム統合前は各事業部で維持・管理されていたが，システム統合後は，図 1 に示した X 社情報セキュリティ基本方針及び対策基準に準拠して管理されることになる。

〔人事管理システムのアクセス制御の設計〕

X 社開発チームから，システム統合の計画作成と実施を依頼された Y 社では，セキュリティエンジニアの H 氏が後輩の G 君を指導しながら，人事管理システムのセキュリティを設計することにした。次は，そのときの G 君と H 氏の会話である。

H 氏：システム統合のために，EL 事業部，SW 事業部及び本社管理部の人事情報を，IT 事業部の RDB に移行することになる。表 1 は，X 社開発チームが設計中の新人事情報データベース（以下，人事情報 DB という）の表のサンプルイメージだ。まずは，何をすべきか検討していこう。

表１　人事情報ＤＢの表のサンプルイメージ（抜粋）

従業員名	性別	従業員番号	事業部/部	グループ	役職	スキル	考課	本給	交通費	…
高橋　美保	女	2002027	IT	PC HW	主任	NW	A	300,000	64,500	…
伊藤　学	男	1989032	EL	パーツ	部長	DB	B	450,000	78,000	…
小林　愛	女	2004024	IT	ディスク	副主任	SU	C	280,000	54,000	…
佐藤　一志	男	2002048	EL	弱電	副主任	－	D	260,000	98,800	…
田中　和子	女	1993002	本社管理部	経理	課長	－	B	360,000	156,000	…
山本　一郎	男	2001028	SW	業務 SW	主任	AD	B	330,000	72,800	…
佐々木　勝	男	1990009	IT	SAN	課長	SV	C	410,000	113,200	…
加藤　千恵	女	1994033	SW	ディスク	主任	AD	B	370,000	84,500	…
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	

G 君：所属事業部の範囲内で検索できるように，人事情報 DB をアクセスするプログラム（以下，業務 AP という）に機能を追加する必要があると思います。

H 氏：そうだね。本社管理部の人事グループでは，事業部ごとに担当者が分かれているので，職務の必要性に応じたアクセス制御ができるようにすべきだね。

G 君：はい。それ以外に，業務 AP の利用者だけでなく，システム管理者も RDB にアクセスすることを考慮して，設計を行う必要があると思います。

H 氏：良い指摘だね。システム管理には様々な業務があるが，例えば，データベース管理とオペレーションの業務を同一の者が実施するのは，セキュリティの観点からは適切な対応とは言えないね。［ a ］の原則を徹底するためには，人事管理システムの①運用体制を含めて考えなければならないな。次に，業務 AP については，どのようなことに注意すべきかね。

G 君：例えば，本社管理部の人事グループの人材管理担当者が，SW 事業部からの要請でセキュリティ技術をもった人材を探す場合には，スキルやグループの情報にアクセスする必要がありますが，給与担当者の場合には，本給や交通費など業務に関係する情報だけを参照できれば十分です。RDB でこのようなアクセス制御を行うには，どんな方法があるのでしょうか。

H 氏：RDB の利用者ビューを効果的に利用すれば，業務 AP への依存度が低いアクセス制御が可能になると思うよ。つまり，RDB を設計するとき，アクセス制御についても十分に考慮しなければならないというわけだ。

G 君：なるほど。業務 AP の利用者が必要とする表，又はその一部の［ b ］だけで構成された利用者ビューを作成するのですね。

H 氏：そのとおり。加えて，X 社の人事管理方針では，例えば各事業部長は自事業部に所属する従業員の情報だけにアクセスできるように制御する必要がある。このような業務要件に対しては，どういう設計をしたらよいか検討してみよう。

G 君：技術的には，表内の［ c ］単位で制御可能な関係データベース管理システム（RDBMS）に切り替えることも選択肢の一つですが，今回のシステム統合では RDBMS は変更しない方針です。あるいは，②RDB の設計でも対応できると思います。この場合，表と利用者ビューは，利

用者の業務要件を満たすように設計します。利用者が人事情報 DB へアクセスする場合には，この利用者ビューだけが利用可能となるように制御すれば，［　a　］の原則が徹底できると思います。

H 氏：そのとおりだね。結論としては，データベース設計にまで踏み込んだ議論が必要なので，X 社開発チームと協議しながら設計を進めていこう。

〔文書管理システムのセキュリティ要件の検討〕

次に，H 氏と G 君は，文書管理システムのセキュリティ要件の実現について技術検討を行うことにした。

G 君：文書管理システムは新製品などの情報も保持することになるので，データの完全性よりも機密性に重点を置いた対策が必要ですね。

H 氏：文書管理システムでは，RDBMS を使わずに，OS ファイルとして情報資産を保持することになる。一般の OS のアクセス制御機能で，X 社の情報セキュリティ対策基準を満たすことが可能かどうか考えてみよう。

G 君：一般の OS では，ファイルに対する操作権限として，参照，書込み，実行などのアクセス制御が可能なので，これらを適切に利用すれば X 社で必要とする機密保護は達成できると考えています。

H 氏：そうだろうか。一般の OS では，ファイルに対してアクセス制御を行っているだけなので，OS のアクセス制御機能だけでは足りないと思うよ。

G 君：具体的には，何が足りないのでしょうか。

H 氏：一般の OS では，基本的にはファイル識別とファイルアクセス者の利用者 ID に基づいてアクセス制御を行う。アクセス制御のルールはアクセス制御リストなどに保持されるが，このリスト管理はファイル所有者が行う。

G 君：ファイル所有者の裁量によってアクセス権が規定されるので，［　d　］アクセス制御と呼ばれているのですね。でも，何が問題なのですか。

H 氏：例えば，あるファイルを別のファイルにコピーするとしよう。元のファイルに対する参照権限と，コピー先ファイルへの書込み権限があれば，ファイルのコピーは可能だ。つまり，ファイルの内容を別のファイルに移動することが可能となる。その結果，［　d　］アクセス制御の場合には，貴重なファイルの内容が意図せず漏えいしてしまう可能性が出てくる。

G 君：分かりました。では，どのような対応が必要なのでしょうか。

H 氏：ISO/IEC 15408 に基づいた，OS 用プロテクションプロファイルの一つである Labeled Security Protection Profile によると，高度な機密保護のためには，［　e　］制御が必要であるとされている。

G 君：なるほど。X 社情報セキュリティ対策基準では，情報が取扱レベルの上位から下位の方向へ移動しないように［　e　］制御されているのですね。

H 氏：そのとおりだ。X 社情報セキュリティ対策基準では，情報資産の機密性を重視するため，取扱レベルが［　f　］以上の情報資産においては［　g　］アクセス制御が必要となるね。

〔文書管理システムのアクセス制御の設計〕

H 氏は，文書管理システム用サーバのパラメタ設定はシステム管理者が行い，システムの起動，停止やデータの移動を含むバックアップ作業はオペレータが行うなど，運用業務の役割を明確に分割した。H 氏は，サーバ側のアクセス制御を徹底するためのシステム構成及び運用体制を X 社開発チームに説明し，了解を得た。

次に H 氏は，X 社開発チームから入手した資料を基に，機密保護対象の情報資産とそれぞれの取扱レベルを表 2 に整理し，アクセス制御の設計作業を進めた。

表２　情報資産と取扱レベル（抜粋）

業務遂行単位	主管グループ	サポートグループ	情報資産名	取扱レベル
A 製品開発プロジェクト	EL 事業部パーツ	－	資料	極秘
B 製品開発プロジェクト	EL 事業部 OEM	IT 事業部ディスク	販売戦略	秘密
			製品仕様書	取扱注
			説明資料	一般
C 製品開発プロジェクト	SW 事業部オフィス	－	企画案	社外秘
D 製品開発プロジェクト	IT 事業部サーバ	SW 事業部 CAD	処理記述	秘密
			DFD	取扱注
			クラス図	社外秘
E 製品開発プロジェクト	IT 事業部 AV	SW 事業部ゲーム	企画案	最高機密
			市場調査結果	秘密
F 製品開発プロジェクト	SW 事業部業務 SW	IT 事業部サーバ	E-R 図	秘密
			画面設計書	社外秘
			処理記述	極秘
健康管理業務	本社管理部総務	－	検診結果	秘密
市場調査業務	EL 事業部営業推進	－	調査結果	社外秘
特許管理業務	SW 事業部知的資産	－	願書	最高機密
財務管理業務	本社管理部経理	－	財務データ	社外秘
⋮	⋮	⋮	⋮	⋮

G 君：表 2 では，文書管理システムの情報資産が業務遂行単位ごとに整理されています。例えば，E 製品開発プロジェクトは IT 事業部 AV グループが主管していて，SW 事業部ゲームグループがサポートグループとして参画しています。このプロジェクトには，“企画案”と“市場調査結果”という 2 種類の情報資産があり，取扱レベルとしてそれぞれ“最高機密”と“秘密”が設定されています。

H 氏：取扱レベルの異なる情報資産が存在する文書管理システムについては，システム構成を考えなければならないな。例えば取扱レベルごとにサーバを割り当てる方法と，複数の取扱レベルの情報

資産を一つのサーバで管理する方法とがある。

G 君：前者は簡単そうですが，X 社情報セキュリティ対策基準においても取扱レベルごとにサーバを分離することまでは要求していないと思いますし，複数の取扱レベルを扱う利用者には使い勝手が悪いと思います。

H 氏：そのとおりだが，X 社情報セキュリティ対策基準では，機密情報資産とそれ以外の情報資産では，扱い方を変えなければならないので，この違いに応じてサーバを分けた方がいいね。

G 君：分かりました。では，文書管理システムを，機密情報資産を管理するサーバ（以下，機密情報管理サーバという）と，それ以外の情報資産を管理するサーバ（以下，管理文書サーバという）の 2 種類で構成することにします。

H 氏：では次に，機密情報理サーバに適用する［ g ］アクセス制御では，機密情報資産の情報区分及び利用者のアクセス区分を表す，ラベルと呼ぶ識別子を使って［ e ］制御を行う。情報資産の情報区分や利用者のアクセス区分に変更が発生すると，ラベルに変更を加える必要が生じる。このラベルの表現形式を考えてくれないか。

G 君：はい。早速ですが，二つのラベルのうち，利用者に付与するラベルについては（L：C）のように表現します。L は利用者の権限クラスを一つ，C はその情報資産へのアクセスを可能とするカテゴリをすべて列記します。この表現形式に従い，X 社情報セキュリティ対策基準に基づいて，利用者のアクセス区分にラベルを付与します。例えば，“クラス 4 ”の権限クラスをもつ利用者は，情報区分中のカテゴリがその利用者のカテゴリと一致した，“秘密”と“社外秘”の機密情報資産を参照できることになります。

H 氏：情報資産の情報区分には，階層関係を示す取扱レベルのほかに，カテゴリもあるので，これも設計してくれないか。

G 君の行ったアクセス制御設計を H 氏がレビューし，内容を検証することになった。

G 君：表 3 は，表 2 を基に機密情報管理サーバに配置する機密情報資産と情報区分の対応を整理したものです。縦軸は取扱レベル，横軸はカテゴリを示しています。業務・プロジェクト名をカテゴリとしました。例えば，“A 製品開発”は A 製品開発プロジェクトを示します。

表３　機密情報管理サーバの機密情報資産と情報区分の対応（抜粋）

カテゴリ ／ 取扱レベル	A 製品開発	B 製品開発	C 製品開発	D 製品開発	E 製品開発	F 製品開発	健康管理	…
最高機密	―	―	―	―	企画案	―	―	…
極秘	資料	―	―	―	―	処理記述	―	…
秘密	―	販売戦略	―	処理記述	市場調査結果	E-R 図	検診結果	…
社外秘	―	―	企画案	クラス図	―	画面設計書	―	…

H 氏：B 製品開発プロジェクトの“販売戦略”に書き込むために必要な利用者のアクセス区分のラベルはどうなるのかな。

G 君：その場合には，（［ h ］：［ i ］）が必要です。

H 氏：D 製品開発プロジェクトと F 製品開発プロジェクトを統括する責任者 P 氏は，情報資産を参照

し，情報資産に修正などが必要であれば，作業担当者に書込みを指示することになる。必要最小限の権限だけを与えるには，どのようなアクセス区分のラベルをもっていればよいのか。

G 君：統括責任者 P 氏には（　j　：(D 製品開発，F 製品開発)）が必要です。

H 氏：例えば，③X 社開発チームのシステムエンジニアである Q 氏が F 製品の“画面設計書”のレビューを実施するとしよう。レビュー時に必要な情報資産にアクセスするためにはアクセス区分のラベルはどうなるのかね。Q 氏に対しては，参照した“画面設計書”に修正コメントを書き込めるようにラベルを付与する必要がある。

G 君：その場合には，（　k　：　l　）が必要になります。

H 氏：分かった。これで，X 社情報セキュリティ対策基準を反映したアクセス制御の基本的な設計はできたようだ。

H 氏は，顧客管理システムについても同様の設計方針で作業を進め，X 社の要求どおりのセキュリティ対策を実現した。

設問１　人事管理システムのアクセス制御の設計について，(1)～(3) に答えよ。

(1) 本文中の　a　～　c　に入れる適切な字句を答えよ。

(2) 本文中の下線①について，どのような体制を採るべきか。65 字以内で具体的に述べよ。

(3) 本文中の下線②の対応について，業務 AP の利用者に対するアクセス制御をどのように設計したらよいか。50 字以内で述べよ。

設問２　文書管理システムのセキュリティ要件の検討について，(1)，(2) に答えよ。

(1) 本文中の　d　～　g　に入れる適切な字句を，　d　及び　g　はそれぞれ 2 字以内，　e　は 5 字以内，　f　は 4 字以内で答えよ。

(2)X 社情報セキュリティ対策基準において，機密情報資産に対するアクセス制御で用いる属性を，利用者 ID 以外に三つ挙げ，それぞれ 5 字以内で答えよ。

設問３　文書管理システムのアクセス制御の設計について，(1)～(4) に答えよ。

(1) 本文中の　h　～　l　に入れる適切な字句を答えよ。

(2) カテゴリを本社管理部又は事業部単位にすると，図 1 に示した情報セキュリティ対策基準に反することになる。その理由を 35 字以内で述べよ。また，その場合，どのようなセキュリティ上の問題が発生するか，60 字以内で具体的に述べよ。

(3) B 製品開発プロジェクトの“製品仕様書”の取扱レベルを“取扱注”から“社外秘”に変更することにした。このときに必要となる一連の二つの処理について，それぞれだれが，どのような処理を行わなければならないか。処理の順に，“～が～する”という形式で，それぞれ 55 字以内で述べよ。

(4) 本文中の下線③の Q 氏に，F 製品の“画面設計書”のレビュー完了後，追加で“処理記述”のレビューも実施してもらう場合，発生すると思われるアクセス制御上の問題と，その解決策を，それぞれ 35 字以内で具体的に述べよ。

問２　安全なデータ転送システムに関する次の記述を読んで，設問 1～4 に答えよ。

J 社は，従業員数 3,000 名規模の自動車エンジン製造会社であり，多くの部品メーカからエンジンの部品を調達している。これらの部品メーカとは，長年にわたって取引を続けている。

J 社では，社内での設計業務や契約業務などが電子化されているので，今後はすべての部品メーカを対象に調達業務の電子化を行うことになった。これまで，部品の設計図や仕様書（以下，調達書類という）は，情報の漏えいや紛失を懸念して書留で郵送していたので，調達業務の電子化を行った場合は，特にデータ転送の安全性を確保する必要があると考えた。

そこで，J 社では，部品メーカとの間で重要なデータを安全かつ確実に転送できるシステム（以下，セキュアなデータ転送システムという）を開発することにした。

〔セキュアなデータ転送システムの要件〕

調達書類は，J 社で作成されて部品メーカに提示される。電子化された調達書類（以下，調達ファイルという）は，データの取扱いに慎重を要するものが多いため，受信した部品メーカが，情報の完全性を確認できる必要がある。また，受信した調達ファイルが，確かに J 社の承認者によって承認されたものであることを，部品メーカが確認できるようにする。

従来は，調達書類を J 社の調達担当者が封筒に入れ，部品メーカの営業部門に書留で郵送していた。部品メーカの営業部門では，郵送されてきた調達書類を営業担当者が受領し，調達書類の内容を確認した後，自社の開発担当者に渡していた。したがって，調達業務を電子化した場合でも，J 社内と部品メーカの内部で想定される脅威については，ほかの手段によって軽減を図ることにして，検討対象としない。

部品メーカは，J 社に比べると小規模であり，専任のシステム管理者を配置できないところが多い。この点を考慮して，部品メーカの調達ファイル受信用の PC にインストールするソフトウェアはなるべく少なくする。

従来の郵送を使った方法に比べて，手順が著しく煩雑になったり，安全性が大きく損なわれたりしないようにする。また，部品メーカの利用者が，特別な講習を受けなくても使えるように，セキュアなデータ転送システムを利用するための専用のアプリケーションを最小限に抑え，導入しやすいシステムにする。

J 社は，これらの要件をシステム開発会社の K 社に提示し，セキュアなデータ転送システムの開発を進めることにした。

〔電子メールの活用〕

K 社では，セキュアなデータ転送システムで使用するアプリケーションについて検討を行った。現在，多くの部品メーカがインターネットを盛んに利用している。特に，電子メール（以下，メールという）は，J 社を含めて部品メーカでも利用者が多いので，メールを利用すれば，セキュアなデータ転送システムの導入コストを抑えられる可能性がある。さらに，用件をメール本文に記載し，調達ファイルを添付ファイルとして，郵送と同様に一括して送信できる。

メールの活用を検討するために，送信者の PC から，送信者の PC と直接通信するサーバまでのデータ転送に，SMTP を使用する方法（以下，SMTP メールという）と HTTP を使用する方法（以下，HTTP

メールという）の二つを取り上げた。図 1 に，SMTP メールと HTTP メールの転送例を示す。

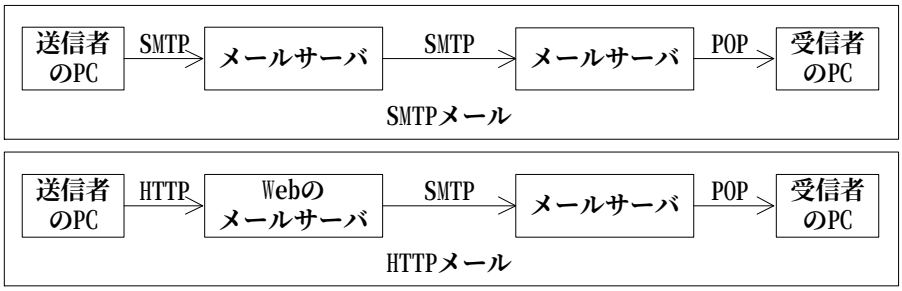

図１　SMTP メールと HTTP メールの転送例

HTTP メールを使用する送信者は，ブラウザを利用してメールの送受信を行うことになるので，SSL で保護した HTTP メールが利用しやすい。SSL は，送信者の電子証明書を用いなくても，Web のメールサーバの電子証明書だけを用いて利用できる。これによって，通信される情報の機密性を確保している。SSL で保護した HTTP メールでは，送信者のパスワードを確認することによって，送信者と Web のメールサーバとの間において，メール送信者の認証ができる。

一方，SMTP メールは，S/MIME が利用しやすい。S/MIME を SMTP メールに適用した場合，メール受信者は，メール送信者を確認できる。また，HTTP メールに比べて，メール送信者と受信者の間で，転送データの漏えい防止や完全性の確保などが容易に行える。

以上の検討結果から，HTTP メールの採用は見送り，SMTP メールに対する S/MIME の適用を検討することにした。

〔S/MIME の検討〕

メールサーバ間の転送でも SMTP が使用されるので，SMTP の安全性を確保する必要がある。SMTP は，ASCII で表現されたテキストデータの転送しかできない。転送される SMTP メールのフォーマットは，RFC 2822 に従う。図 2 に，S/MIME を使用した SMTP メールのフォーマットの概要を示す。

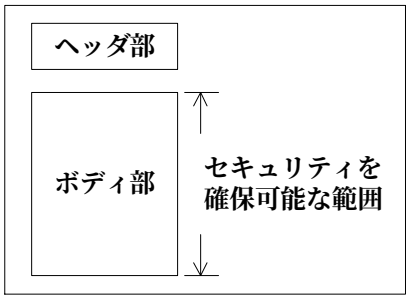

図２　S/MIME を使用した SMTP メールのフォーマットの概要

S/MIME は，メールの送信者と受信者の間で公開鍵基盤（以下，PKI という）を利用して，送信者が付与した［　a　］を受信者が検証することによって，メールとして転送されるデータの完全性の確保と［　b　］の確認ができる。さらに，PKI は，S/MIME においてメールとして転送されるデータ

の機密性を確保するためにも利用される。ここで用いられる非対称暗号（RSA 暗号）では，秘密鍵と公開鍵が 1 対 1 に対応している。ある秘密鍵を使用して [c] した情報は，[d] と対を成す [e] で復号できる。また，[e] を使用して暗号化した場合，[d] で復号できる。一般に，非対称暗号は暗号化するデータ量が多くなると，対称暗号に比べて処理時間が長くなる。効率性の観点から，S/MIME では，メールの本文を対称暗号で暗号化し，この暗号化に使用した共通鍵を，非対称暗号で暗号化する。この仕組みによって，送信者が指定した受信者だけがメールを読むことができるので，データ転送の安全性が高まる。

S/MIME を使用した場合，添付されるバイナリデータは，ASCII で表現される 64 種類の文字と特殊用途の“ = ”を使って表現する [f] 方式に従って，テキストデータに変換される。64 種類の情報を表現するために必要なビット数は，[g] ビットである。ここで，1 バイトが 8 ビットで，[g] と 8 の最小公倍数が [h] になり，[h] ビットは [i] バイトである。したがって，[i] バイトのバイナリデータから，4 バイトのテキストデータが生成される。つまり，添付されるテキストデータは，パディング文字や改行文字などを無視すると，バイナリデータの [j] 倍になる。このことから調達ファイルのような大量のデータを転送する場合には注意が必要である。

〔Web サーバの併用〕

K 社は，S/MIME による転送データ量の増加への対策として，調達ファイルをメールに添付するのではなく，J 社が運用する Web サーバからダウンロードしてもらう方法を採用することにした。具体的には，J 社は部品メーカに対して，S/MIME を使用した SMTP メールに，ダウンロードするための情報を記載して送信する。この情報は，Web サーバの URL やダウンロードされるファイルを指定する文字列（以下，ファイル指定文字列という），部品メーカごとに割り当てられたダウンロードキーなどから構成される。

ダウンロードされるファイルは，調達ファイルと，その完全性の確保や承認者の確認などを目的として添付した，J 社の承認者の [a] などから構成される。J 社は，Web サーバの併用に伴い，ダウンロードしたファイルの [a] をチェックしたり，調達ファイルを抽出したりするクライアントソフトウェア（以下，クライアント AP という）を，部品メーカに提供することにした。

部品メーカでは，営業担当者が J 社から S/MIME を使用した SMTP メールを受信し，そのメールの送信者が J 社の調達担当者であることを十分に確認する。その後，開発担当者の業務状況や調達内容から，調達に応じるのに適任の開発担当者を選び，当該メールの内容を転送し，調達ファイルを取得するように指示する。このとき，営業担当者は，S/MIME を使用せずに開発担当者に SMTP メールを転送するので，J 社からの S/MIME を使用した SMTP メールを読むための電子証明書は，部品メーカの営業担当者だけが所有していればよいことになる。

指示を受けた開発担当者は，自ら PC を操作してファイルをダウンロードする。図 3 に，ファイルをダウンロードする操作手順の概要を示す。

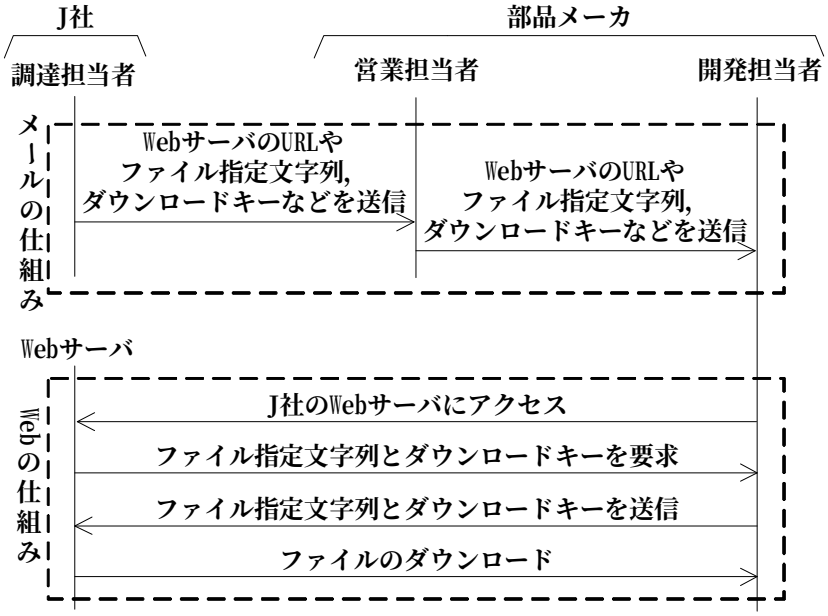

図３　ファイルをダウンロードする操作手順の概要

開発担当者が，J 社から割り当てられた利用者 ID とパスワードとを使用して，Web サーバにログインした後，営業担当者からのメールで指示された URL に従ってダウンロードする Web ページにアクセスする。開発担当者のパスワードは，セキュアなデータ転送システムの安全性を考慮して，ある程度の期間で更新していくことにした。当該ページにアクセスすると，ファイル指定文字列とダウンロードキーの入力を要求する Web ページが表示される。開発担当者は，メールで指定された値を入力し，“ダウンロードボタン”を押してファイルをダウンロードする。Web サーバは，開発担当者の電子証明書を使用せず，自らの電子証明書だけを使用した SSL を利用し，ダウンロードキーの検証を行う。

〔Web サーバのセキュリティ〕

K 社において，J 社の Web サーバで動作するソフトウェアの開発を行っている T 君は，リーダの S 氏に相談しながらセキュアなデータ転送システムの具体的な検討を進めた。T 君は，Web サーバの安全性を確保するために，部品メーカの開発担当者が，許可されていないファイルをダウンロードすることを禁止する方法を考えた。

Web サーバにおいて，あるディレクトリの配下に，各部品メーカに対応したディレクトリを用意し，ダウンロードされるファイルは，その配下に格納される。ダウンロードされるファイルが格納されているディレクトリ名には，各部品メーカに対してあらかじめ割り当てた名称を使用し，開発担当者には通知しないことにした。また，開発担当者が入力するファイル指定文字列と，入力したファイル指定文字列に基づいて Web サーバで認識されるファイル名を異なるものとした。つまり，開発担当者には，Web サーバで認識できるディレクトリ名やファイル名を直接指定させない。

T 君は，各部品メーカの開発担当者には，メールで通知されたダウンロードキーとファイル指定文字列だけを入力してもらい，ダウンロードキーと，Web サーバで該当するファイルが格納されているディレクトリを対応させた。ダウンロードキーが部品メーカごとに設定されていることから，Web サーバで認証された部品メーカの開発担当者に対して，該当するディレクトリ配下のファイルだけをアクセスさ

せることができる。

ダウンロードされるファイルのファイル名は，ASCII で表される数字，英小文字，“(”，“)”，“#”，拡張子を区切るためのピリオドの 40 文字を用いて表現する。T 君は，調達ファイルのデータ量が大きいことから，データ転送量を抑えるために，ダウンロードする前にファイル圧縮技術を適用したいと考えた。そこで，ダウンロードされるファイルの内容は，暗号化せずに，一般に広く使用されている圧縮したり伸張したりするアプリケーションプログラム（以下，CEP という）を利用して，あらかじめ圧縮しておくことにした。この場合，ファイルを CEP で圧縮しているので，Web サーバで認識できるファイル名は，開発担当者が入力したファイル指定文字列に，“.”と拡張子“cep”を付加したものとする。

T 君は，これらの仕組みによって，開発担当者が Web サーバで認識されるファイル名を直接入力できないようにして，Web サーバの安全性を確保しようとした。T 君は，部品メーカの開発担当者からダウンロード要求を受け付けて処理を行う CGI プログラムを，Perl を用いて作成し，セキュリティに関して S 氏のレビューを受けた。図 4 に，CGI プログラム（抜粋）を示す。

```
#1/usr/bin/perl

sub return_cep_file {
   if (open FILE, $_[0]) {
      # ファイルの内容をそのまま出力し，FILE をクローズ
   } else {
      # エラー処理は省略
   }
}

$buffer = $ENV{'QUERY_STRING'};
# ブラウザから送られてきた引数を$buffer に格納

($arg1, $arg2) = split(/&/, $buffer);
($prm, $fname) = split(/=/, $arg1);
# ファイル指定文字列を$fname に格納

$fname =~ s/%([a-fA-F0-9][a-fA-F0-9])/chr(hex($1))/eg;
# $fname の URL エンコーディングをデコード

($prm, $d_key) = split(/=/, $arq2);
# ダウンロードキーを$d_key に格納

$parts_dir = check_d_key( $d_key );
# check_d_key は，ダウンロードキーを検証し，
# 正当性を確認できたら，部品メーカに対応したディレクトリ名を返却
# 正当性を確認できなかったら，空白文字列を返却

if($parts_dir){
    return_cep_file( $parts_dir.'/'.$fname.'.cep' );
}else{
# エラー処理は省略
}
```

図４　ＣＧＩプログラム（抜粋）

T 君：ファイル指定文字列やダウンロードキーを入力したり，ファイルをダウンロードしたりする Web サーバとの通信には，すべて SSL を使用して安全性を確保します。

S 氏：そうですね。各部品メーカに対応したダウンロードキーは，J 社から部品メーカに通知されています。このダウンロードキーをブラウザから Web サーバに送る必要があります。したがって，これらの間の通信の安全性確保は重要です。さらに部品メーカごとに対応したダウンロードキーを検証するために使用する，ダウンロードキーを記録したファイルなど，重要なファイルが外部

に漏えいしないように注意する必要があります。

T 君：認証された部品メーカの開発担当者は，ディレクトリ名を知らないので，ダウンロードを許可されているファイル以外にはアクセスが困難です。

S 氏：ファイル指定文字列の指定方法を教えてください。

T 君：ファイル指定文字列は，例えば，部品メーカの開発担当者に“sample”と入力してもらい，CGI プログラムにおいて“sample.cep”のように拡張子を付加して，Web サーバ内で認識可能なファイル名を生成しています。

S 氏：図 4 では，正しいダウンロードキーを入力した部品メーカの開発担当者が，自社に割り当てられていないディレクトリ配下にあるファイルをダウンロードできる危険性があります。開発担当者は，自らの PC で動作するブラウザでアクセスするので，任意の文字列を入力できるのです。

T 君：部品メーカの開発担当者が，自社に割り当てられていないディレクトリにアクセスできる危険性は分かりました。しかし，CGI プログラムでは，Web サーバで認識できるファイル名の拡張子を“cep”に限定しているので，ファイル名の拡張子が“cep”以外のファイルには，アクセスできません。

S 氏：そうとも言えないでしょう。Perl の文字列の処理方法を十分に調査して，再度検討してみてください。

T 君は，①S 氏の指摘を受けて，部品メーカの開発担当者が Web サーバの自社に割り当てられていないディレクトリをアクセスしたり，Web サーバ内の任意のファイル名を入力したりできないように，CGI プログラムを改良した。

K 社は，S/MIME を使用した SMTP メールと Web の仕組みを利用した，セキュアなデータ転送システムを完成させて J 社に納入した。J 社は，このシステムの運用を開始し，調達業務の電子化を完了した。

設問１　SSL を使用した HTTP メールについて，(1)，(2) に答えよ。

(1)　図 1 で示した範囲において，情報漏えいの防止が困難な場所を二つ挙げ，それぞれ 15 字以内で述べよ。

(2)　メール送信者の否認防止に効果がある場所を特定し，その理由を含めて，30 字以内で述べよ。

設問２　S/MIME を使用した SMTP メールについて，(1)～(3) に答えよ。

(1)　本文中の［　a　］～［　j　］に入れる適切な字句を答えよ。
なお，［　j　］は，小数第 2 位を四捨五入して小数第 1 位まで求めよ。

(2)　情報漏えいの防止のために，subject フィールドに記述する内容に注意する必要がある。その理由を，20 字以内で述べよ。

(3)　部品メーカの営業担当者が，J 社の調達担当者から送信されるメールの送信者を確認する方法を，40 字以内で具体的に述べよ。

設問３　Web サーバのセキュリティについて，(1)～(3) に答えよ。

(1) 部品メーカが開発担当者のパスワードを管理する方式に比べて，J 社が開発担当者のパスワードを部品メーカに通知する方式の方が有用である。その利点を，30 字以内で述べよ。

(2) 図 4 で示した CGI プログラムを使用した場合，自社に割り当てられていないディレクトリをアクセスする方法と，任意のファイルをアクセスする方法を，それぞれ 35 字以内で述べよ。

(3) T 君が，本文中の下線①で示した S 氏の指摘を受けて，図 4 で示した CGI プログラムに追加すべき機能を，55 字以内で述べよ。

設問４　Web サーバを用いたファイルのダウンロードについて，(1)～(3) に答えよ。

(1) データ転送の安全性を確保するために SSL を利用した理由を，Web サーバでのファイルの保存状態に着目して，30 字以内で述べよ。

(2) Web サーバでの操作において，ダウンロードキーを利用した理由を，部品メーカにおける開発担当者の選任方法に着目して，30 字以内で述べよ。

(3) 部品メーカの開発担当者が，Web サーバにアクセスしたとき，最初に確認すべきことを，20 字以内で述べよ。また，その目的を，50 字以内で述べよ。

プログラム言語 Perl の用例・解説

Perl を使用した問題では，各問題文中に注記がない限り，次に示す用例に従って記述する。
なお，用例は，解答で使用する演算子，関数，予約語などを制限するものではない。

種類	用例 解説

1.注釈

#	#ここにコメントを書く 行末までが注釈となる。

2.リテラル

スカラ	123 10 進数 123 である。
	12.3 10 進数 12.3 である。
	4E-5 10 進数 4×10^{-5} である。
	0x9f 16 進数 9F である。
	0147 8 進数 147 である。
	0b010111 2 進数 010111 である。
	$var = "hello"; print '$var ', "$var ",`echo world`; 変数 var に文字列“ hello ”を代入する。文字列のスカラ'$var ', "$var ", `echo world` を出力する。"$var "は変数を展開し，`echo world`はコマンドの出力を展開するので，出力は“ $var hello world ”となる。
	＼n 制御文字（改行）である。
	＼r 制御文字（復帰）である。
	＼t 制御文字（水平タブ）である。

リストリテラル	('a', 'b', 'c')
	リスト('a', 'b', 'c')である。
	('a', 'b', 'c')[0]
	リスト('a', 'b', 'c')の１番目の要素'a'である。
	()
	空リストである。
	('a' => 'alpha', 'b' => 'bravo', 'c' => 'charlie')
	キーa，b，c に，それぞれ値 alpha，bravo，charlie を結び付けたハッシュである。
ファイルハンドル	STDIN
	標準入力である。
	STDOUT
	標準出力である。
	STDERR
	標準エラー出力である。
	AV
	コマンドラインから指定されたファイル名のリストを順に読み込むためのファイルハンドルである。

3.変数

スカラ変数	$var
	スカラ変数 var である。
配列変数	@ary
	配列変数 ary である。
配列要素	$ary[6]
	配列変数 ary の 7 番目の要素である。
ハッシュ変数	%hash
	ハッシュ変数 hash である。
ハッシュ要素	$hash{'a'}
	ハッシュ変数 hash の要素のうち，キーa に結び付けられた値である。
局所的な変数	{my $var;}
	{}内を有効範囲とする変数 var の宣言である。
$_	$_ = "abc"; if (/b/) print "match";
	パターンマッチの演算子が省略されたとき，$_の文字列“ abc ”が//内のパターン b と一致するかどうかを判定し，“ match ”が出力される。
@ARGV	@ARGV
	コマンドライン引数のリストを格納する配列変数である。
@_	@_
	サブルーチンに渡す引数のリストを格納する配列変数である。

4.演算子

->	$object->method1
	オブジェクト object のメソッド method1 を呼び出す。
	Class->method2
	クラス Class のメソッド method2 を呼び出す。
++, --	$a++
	変数 a を評価した後に 1 を加算する。
	--$b
	変数 b から 1 を減算した後に評価する。
!, +（単項), -（単項）	!$a
	変数 a の論理否定である。
	+123
	正の数 123 である。
	-123
	負の数 123 である。
=~, !~	$html_contents =~ /<b>/
	変数 html_contents の値に，文字列“<b>”が含まれているときに真を返す。
	$html_contents !~ / /
	変数 html_contents の値に，文字列“ ”が含まれていないときに真を返す。
*, /, %	314 * 34
	314 と 34 の乗算である。
	6 / 469
	6 を 469 で割る除算である。
	34 % 6
	34 を 6 で割る剰余演算である。
+, -, .	3.14 + 2.72
	3.14 と 2.72 の加算である。
	220-8125
	220 から 8125 を引く減算である。
	"IPA"."JITEC"
	文字列“IPA”と“JITEC”の連結である。
<, >, <=, >=, lt, gt, le, ge	1 < 2
	数値 1 と 2 を比較し，演算子の左側が右側より小さいので真を返す。数値の関係演算子には，ほかに>，<=，>=がある。
	"b" lt "a"
	文字列“b”と“a” を比較し，演算子の左側が右側より小さくないので偽を返す。文字列の関係演算子には，ほかに gt，le，ge がある。
==, !=, <=>, eq, ne, cmp	1 <=> 2
	数値 1 と 2 を比較し，演算子の左側が右側より大きければ 1，等しければ 0，小さければ－1 を返すので，この場合は－1 を返す。数値の比較演算子には，ほかに==，!=がある。
	"b"　cmp　"a"

	文字列“b”と“a”を比較し，演算子の左側が右側より大きければ1，等しければ0，小さければ−1 を返すので，この場合は 1 を返す。文字列の比較演算子には，ほかに eq，ne がある。
&&	$x >= 0 && $x < 10 変数 x の値が 0 以上かつ 10 未満なら真を返す。
||	$x < 0 || $x >= 10 変数 x の値が 0 未満又は 10 以上なら真を返す。
..	@card = (1 .. 52) 1 から 52 までの連続する整数を配列変数 card に代入する。
=，+=，-=， *=，/=，%=	$a = 1 変数 a に 1 を代入する。
	$a += 10 変数 a の値に 10 を加算して a に代入する。 代入演算子には，ほかに-=，*=，/=，%=がある。
=>，，	%hash = ('a' => 'alpha', 'b' => 'bravo', 'c' => 'charlie') a に alpha，b に bravo，c に charlie を結び付けたハッシュをハッシュ変数 hash に代入する。
not	not $a 変数 a の論理否定である。
and	$a < 0 and $b == 0 変数 a が 0 より小さいか，変数 b が 0 と等しいかという二つの関係式の論理積である。
or， xor	$a < 0 or $b == 0 変数 a が 0 より小さいか，変数 b が 0 と等しいかという二つの関係式の論理和である。
	$a < 0 xor $b == 0 変数 a が 0 より小さいか，変数 b が 0 と等しいかという二つの関係式の排他的論理和である。

注　演算の優先順位は，上表の枠の順である。

5.文

if	if ($var == 1) {
	　print "a";
	} elsif ($var == 2) {
	　print "b";
	} else {
	　print "c";
	}
	変数 var の値が 1 なら“a”を，2 なら“b”を，それ以外なら“c”を出力する。

while	$i = 1;
	while($i <= 10) {
	　print $i++, "\n";
	}

	変数 i の値を 1 から 1 ずつ増やし，10 回出力する。

for	for($i = 1; $i <= 10; $i++){
	　print "$i\n"1;
	}

	変数 i の値を 1 から 1 ずつ増やし，10 回出力する。

foreach	foreach $i (1, 3, 5){
	　print "$i\n";
	}

	変数 i にリストの各要素 1，3，5 を順に代入し，3 回出力する。

next	for ($i = 1; $i <= 10; $i++) {
	　next if $i % 2;
	　print "$i\n";
	}

	変数 i が 2 で割り切れないとき，ループ本体の next 行より後を実行しないので，偶数を出力する。

6.正規表現

\	/\. \^\$\[|\+*\?\{\(\)\/\\/
	次の 1 文字そのものを表す。“ .^$[|+*?{()/ \ ”と一致する。
.	/www.ipa.go.jp/
	改行文字以外の任意の 1 文字と一致する。“ wwwdipa,go@jp ”と一致する。
^	/^ab/
	先頭が“ ab ”である文字列と一致する。“ abc ”と一致するが，“ cab ”とは一致しない。
$	/yz$/
	末尾が"yz"である文字列と一致する。“ xyz ”と一致するが，” yza ”とは一致しない。
+	/qo+d/
	直前の 1 文字 o の 1 回以上の繰返しと一致する。“ god ”や“ gooood ”と一致するが，“ gd ”とは一致しない。
*	/go*d/
	直前の 1 文字 o の 0 回以上の繰返しと一致する。“ gd ”，“ god ”や“ gooood ”と一致する。
?	/colou?r/
	直前の 1 文字 u の 0 回又は 1 回の出現と一致する。“ color ”又は“ colour ”と一致する。

{m}, {m,n}	/co{2}l/
	直前の 1 文字 o の 2 回の繰返しと一致する。“cool”と一致するが，“col”や“coool”とは一致しない。
	/go{1, 3}d/
	直前の 1 文字 o の 1～3 回の繰返しと一致する。“qod”や“qood”と一致するが，“gd”や“gooood”とは一致しない。
(…)	/<(h.)>/
	()内の文字列と一致するパターンを部分パターンとしてまとめる。“<h1>”と一致した場合は“h1”が，“<hr>”と一致した場合は“hr”が，まとめられる。
＼1, ＼2, …	/<(..)><([bp])>JITEC<＼/＼2><＼/＼1>/
	左から順に()内のパターンと一致した文字列が＼1, ＼2, …に割り当てられる。“ JITEC ”と一致するが，“<td>JITEC</P></td>”とは一致しない。
[…]	/<h[12r]>/
	[]内で指定した文字 1，2 又は r のどれか一つと一致する。“<h1>”，“<hr>”と一致するが，“<h3>”や“<HR>”とは一致しない。
	/[^0-9]/
	[]内で指定した 0～9 以外の 1 文字と一致する。“a”と一致するが，“3”とは一致しない。
…|…	/<(a href|img src)=/
	|で区切られた“a href”又は“img src”のどちらか一方と一致する。 “<a href=”や“<img src=”と一致するが，“<A HREF=”や“<img height=”とは一致しない。

7.サブルーチン

定義	sub greeting{
	　print "hello Perl＼n";
	}

	“hello Perl”を出力するサブルーチン greeting を定義する。
呼出し	subroutine ($arg1, $arg2);
	サブルーチン subroutine を引数 arg1 と arg2 で呼び出す。()を省略して“subroutine $arg1, $arg2; ”とする表記もある。
戻り	return －1;
	サブルーチンから抜け出し，値－1 を返す。

8.モジュール

use	use CGI;
	モジュール CGI を 1 度だけ読み込み，利用可能にする。

9.メソッド呼出し

->	$object->method1(arg1); 演算子 ->を使って，オブジェクト object のメソッド method1 を引数 arg1 で実行する。
	Class->method2(arg1, arg2); 演算子->を使って，クラス Class のメソッド method2 を引数 arg1 及び arg2 で実行する。

10.文字列操作関数

chomp	chomp @lines; 配列変数 lines の各要素の末尾にある改行文字を削除する。
eval	eval $exp_str; 変数 exp_str の内容を Perl プログラムとして解釈し実行する。
length	length $long_str; 変数 long_str に格納される文字列の文字数を返す。

11.配列・ハッシュ操作関数

keys	%hash = ('a' => 'alpha', 'b' => 'bravo', 'c' => 'charlie');
	foreach $key (keys %hash){
	　print "$key＼n";
	}
	ハッシュ変数 hash のキーのリストを取り出し，各キーを出力する。この場合は，“a”,“b”,“c”を順不同に出力する

shift	$next = shift @queue; 配列変数 queue の先頭要素を取り除いて詰め，取り除いた値を変数 next に代入する。
sort	@pile = sort @jumble; 配列変数 jumble の値を文字列の大小比較によって昇順に整列し，配列変数 pile に代入する。
	@pile = sort {$b <=> $a} @jumble; 配列変数 jumble の値を数値の大小比較に従って降順に整列し，配列変数 pile に代入する。
split	@fields = split ',',$csv; 変数 csv の値をコンマで区切って分割したリストを配列変数 fields に代入する。

12.検索・置換関数

M/…/又は /…/	$html_contents =~ /<b>/i; 変数 html_contents の値が，文字列“<b>”又は“<B>”を含んでいるかどうかを判定する。i は，大文字，小文字の区別をしないオプションである。
S/…/…/	$html_contents =~ s/ /＼n/gi; 変数 html_contents の中の文字列“ ”，“ ”，“ ”又は“ ”を改行文字に置換する。g は，一致したすべての文字列を置換するオプションである。
$`, $&, $', $1, $2, …	'The date is 1970-01-23.' =~ /([0-9]{4})-([0-9]{2})-([0-9]{2})/; print "String before the date: $`＼n"; print "Date: $&＼n"; print "String after the date: $'＼n"; print "Year: $1＼n", "Month: $2＼n", "Day: $3＼n"; 文字列“The date is 1970-01-23. ”に対して，一致した部分の前の文字列，一致した文字列，一致した部分の後ろの文字列をそれぞれ変数 `, &, ' に代入する。また，()で囲まれた部分パターンと一致した文字列を，１番目から順に変数 1，2，3 に代入する。これらを利用し，“String before the date:The date is”，“ Date:1970-01-23 ”，“ String after the date:. ”，“ Year:1970 ”，“ Month:01 ”，“ Day:23 ”の６行を出力する。

13.入出力操作関数

open	open LOG, '>>cgi.log'; ファイル cgi.log を追記モードで開き，ファイルハンドル LOG に対応付ける。
<*filehandle*>	$line = <USER_FILE>; ファイルハンドル USER_FILE から１行を読み込んで変数 line に代入する。
<>	@records = <>; 標準入力（コマンドライン引数があるときは，コマンドライン引数で指定されたファイル）から順にデータを読み込み，すべての行を配列変数 records に代入する。
print	Print LOG "sync.＼n"; ファイルハンドル LOG に対応するファイルに文字列を出力する。
close	close LOG; ファイルハンドル LOG に対応するファイルを閉じる。

14.システムインタフェース

die	open(FILE, 'a_file')or die 'cannot open a_file'; ファイル a_file を開く。開くのに失敗したとき，“cannot open a_file”というメッセージを出力して実行を終了する。
system	system 'a.out'; コマンド a.out を実行し，コマンドが終了するまで待機する。