演習問題 解答例

1章 1-4 演習問題（P.11）

（1）オペレーティングシステムの役割と二つの機能との関係を明らかにせよ．

［解答］

OSの役割は使いやすさの提供，信頼性の提供，性能保証機能の三つである．そしてOSの主機能は機能マシンの提供と資源管理である．図1･2に示したとおり，機能マシンの提供はアプリケーションプログラムを開発するプログラマに対するコンピュータの使いやすさを提供している．またコンピュータシステムを構築する際の性能設計，信頼性設計などにもOSの役割として備えられた機能を利用できるようになっている．一方，資源管理機能はシステム運用・管理者に主として提供された機能であり，OSの備えるハードウェアの障害監視，性能情報の提供，などがある．

（2）PCとサーバマシンとのオペレーティングシステムに求められる機能の相違点を論ぜよ．OSの主機能との関係において説明せよ．

［解答］

OSの役割である使いやすさ，信頼性と性能保証の各側面から考察する必要がある．PCでは使いやすさが第1目標となるが，企業活動に重要な使命を担うサーバや社会のインフラストラクチャとなっているサーバは信頼性，性能などが第1目標となるはずである．OSの二つの主機能は機能マシンの提供と資源管理機能であるが上記の理由から資源管理機能（非機能要件とも呼ばれる）が重要な要素となる．

（3）PCで多重プログラミングが利用できないときの不都合は何か.

［解答］

PCにおいて多くのエンドユーザはメール，Webブラウザなどを常時使用し，また文書作成や表計算ソフトなどを同時に利用するのが一般的である. このため，多重プログラミングが不可能なOSではアプリケーションプログラムを同時に立ち上げておくことができないことになり，PCの利便性が失われる．この理由により多重プログラミングは今やPCのOSにとって必須の条件となっている.

（4）PCでマルチプロセッシング（デュオやクワッドなど）を使う具体的な利点は何か.

［解答］

PCでは多重プログラミングの利用を前提にした使用形態が一般的になっている．例えば，文書作成プログラムを使用中にもメールソフトはメールが送られてきているか否かを定期的にメールサーバに問い合わせできることが望ましい. Webブラウザで動画を再生している場合などは大量のパケットを受け取る必要があり割込みが頻発する．割込みによるパケット受取りと動画再生は並列して実行する必要がある．このように多重プログラミング環境でCPUを多用するアプリケーションが複数同時に実行する場面があるため，複数のCPUを同時に実行するハードウェアは処理能力を満たす利点がある.

2章　2-4 演習問題（P.28）

（1）割込みマスクを設定することでOSは外部割込みを抑止可能となる．そこで表2･1にある入出力完了，タイマー切れ，外部信号入力，ハードウェアエラーが同時に発生したとき，どの順に割込み処理すべきだろうか．その順位付けをし，その理由を説明せよ.

［解答］

OSの設計はその適用分野により方針が様々である．割込み優先順位の決定は特に決まったルールはない．一般的にはハードウェアのエラーに対する対処は優

先度を高くすべきである．ハードウェアのどのような障害が割込みになるかはシステムによって様々であるため一概に扱うことはできない．リアルタイム性の要求が強い応用分野のOSの場合は，タイマー切れの割込み優先度は高くすべきであろうが，タイマーに対する割込み要求が秒単位の場合であるならば優先順位は低くても許されるかもしれない．制御系のコントロールが主である場合で外部信号に対するリアルタイム性が要求される場合は割込み優先度をほかよりも高くすべきだ．以上の観点から入出力完了の割込み順位は自ずと定まるはずである．

（2）磁気ディスクの回転性能が7200rpm（rotations per minute）である場合平均サーチ時間を計算せよ．

［解答］

ディスクの一回転に要する時間は：

60〔sec〕/7200〔rotate〕= 1/120〔sec/rotate〕≒ 8.3〔ms/rotate〕

である．平均サーチ時間なので，1/2回転待ちと解釈し，この1/2の時間：

8.3/2 ≒ 4.2 ms

である．

（3）図2･9は入出力ソフトウェアの階層が示されている．このように階層的なソフトウェア構成の利点について論ぜよ．

［解答］

階層（レイヤー）間のインタフェースの明確化が可能．レイヤーソフトはインタフェースだけを守ればどのようなアルゴリズムで機能を実現してもよく，このためソフトウェアの開発が部品作成のように独立して自由に行える利点がある．また，インタフェースが明確なのでバグの所在が明確化できるなどの利点がある．

（4）本章2-3-7では磁気ディスクアクセス時間の説明をしたが，図2･13に示すようにシーク時間が最も長いならばこの時間の短縮が重要である．そこで図2･10のようにある磁気ディスクへの入出力要求が重なっていたときにシーク時間を短縮するスケジューリング方法を考案し，利点・欠点を説明せよ．

［解答］

(a)現在のヘッド位置に最も近い要求から処理する方法，(b)入出力要求のヘッドを（番号の昇順，降順方向などの）一方向にサービスするエレベーターアルゴリズム，(c)FIFOアルゴリズムなどが考えられる．利点は特徴から明らか．欠点は(a)現ヘッド位置から遠い要求は待ち時間が想像以上に長くなる可能性があること，(b)トラック番号の最小・最大に近い要求は待ち時間が長くなる可能性があること，(c)ディスクの全体としての使用効率は改善されない，など．

（5）図2･7にはメモリ保護機構が説明されている．メモリにはページ単位に読出し，書込み，命令実行の属性がある．これらは禁止か許可かを示しているので全部で8種類の属性となる．ではこれらの中で意味のある属性と使用目的を述べよ．

［解答］

読出し(r)，書込み(w)，命令実行(x)と略しページの属性を(r,w,x)で表す．(r,-,-)は読出し可能だけを意味する．(-,-,-)は明らかに意味のない属性である．このようにして考えよ．

（6）メモリには空き領域がたくさんあり，多くのプロセスが入出力の処理完了を待っている状態にある．そして，CPUの利用率が30％であるときこのコンピュータの運用はどのようにすべきか．

［解答］

CPUとメモリの利用率は低く入出力がボトルネック（隘路）となったアンバランスな状態にある．入出力装置を増強しCPU利用率を上げ，またマルチプログラミングの多重度向上を図るバランスのよい運用とする必要がある．

（7）OSが割込み禁止で実行しなくてはならない場合はいかなるときだろうか．OSがある割込み処理中にほかの割込みを許す場合には，何を注意すべきだろうか．

［解答］

割込み処理の初期段階は割込み情報の収集，ならびにその後の処理の準備など

がある．この処理の間に割込みが発生すると処理に矛盾が生じてしまうため割込み禁止とせざるを得ない．その後，カーネル実行中に割込み受付け可能とするには，カーネルは再帰的（recursive）に実行可能なプログラム構造となっていなければならない．このようなカーネルの構造に設計すれば，カーネル自身がシステムコールを実行しカーネル自身のサービスを受けることができる．

（8）コンピュータが割込み禁止状態でかつ命令をフェッチ（読出し）しない状態とはいかなるときか説明せよ．

［解答］

割込み禁止（disable）で命令をフェッチしない状態（wait）になると，その状態解除は不可能となる．つまり，完全なシステムダウンである．disable-wait状態という．カーネルが自身の論理的矛盾を判断し，動作不能状態と判断したとき，故意にdisable-waitとし，矛盾を判断した根拠をコードとしてコンソールなどに表示し停止することがある．UNIX系OSでのカーネルパニックはその一種である．

（9）OSは実行すべきプロセスがないとき，例えばすべてのプロセスが入出力待ち状態になっているなど，何をすべきか．

［解答］

CPUを使うプロセスがないときOSはアイドル状態にする．具体的には命令実行を停止し何らかの外部割込みを待つ状態とする．命令停止ができないマシンでは小さな無限ループを実行して割込みを待つ．IA32ではhalt命令により命令フェッチを停止する．この命令は特権命令である．このような状態はenable-waitということがある．

3章　3-7 演習問題（P.75）

（1）図3･13では順編成ファイルの先読み効果を説明しているがシステムバッファが一杯になったときに追い出すべきブロックのアルゴリズムならびに具体

的方法を考えよ．

［解答］

最も昔に読み込んだブロックから追い出す方法（LRU：Least Recently Used）は典型的な置換アルゴリズムである．

（2）UNIX系のファイルは本章3-5-2で述べたようにファイルは一次元のバイト列でありレコードのような構造はなく，ファイルを扱うプログラムが意識すればよいという考えに立っている．この利点と欠点について考察せよ．

［解答］

ファイル管理は非常に単純になる．目的のデータは先頭からの相対バイトアドレスで求められる．その長さもプログラムが指定すればよい．したがってデータ構造はすべてプログラマの設計責任となるのでデータ構造を意識したプログラミングが必要とされる．

（3）図3･25にはi-nodeとディレクトリファイルを参照してファイル探索の方法が示されている．この方法では二分されている情報を交互に参照する必要があり入出力回数が多くなり性能面での問題が生じかねない．そこでこの方式を改善する方策を考えよ．

［解答］

i-nodeは容量が比較的小さく参照が多いデータである．このためキャッシュメモリ領域を用意して入出力の削減を図ることが期待できる．また，ホームディレクトリやルートディレクトリなども同様にキャッシュメモリを用意することで入出力を削減する．

（4）OSがハードウェアを操作する特権命令でファイルに対する読み書きを起動するためにread, writeがシステムコールになっている．ではopenがなぜシステムコールである必要があるのだろうか．

［解答］

ファイルへの読み書きを実行する前に，各々のファイルへのアクセス権限をチェックする必要がある．このためアクセス権のチェックをカーネルが実行するた

めに特権状態である必要がありシステムコールインタフェースを通す必要がある.

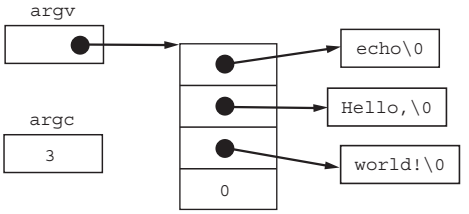
(5) 図3･37にはルートディレクトリを読むプログラム例がある．そこで以下の発展的拡張を試みよ．付録Bを参考にしてもよい.

(a) ホームディレクトリを読むプログラムにする.

(b) mainの引数(int argc, char *argv[ ])を使いプログラム実行時に引数としてディレクトリ名を与える方法に拡張する.

(c) stat/fstatなどを用いてファイル情報を極力詳しく出力する.

Cの実行環境ではmainが呼び出されると図のような引数がプログラムに渡される.



“$./echo Hello, world!”を入力した際に
main(int argc, char* argv[ ]) に渡されるパラメータ

[解答]

付録B 参照

(6) ファイルを操作するシステムコールの実際を試みる.＜付録C参考＞

(a) ファイルを生成（creat）し端末から文字を入力し（read）それらをファイル内にテキストレコードとして格納（write）するプログラムを作成（close）してみよ.

(b) 上記のファイルを表示するプログラムを作成してみよ（open, read, stdoutへの出力など).

(c) このファイルを更新するプログラムを作成してみよ.

[解答]

付録C 参照

4章　4-5 演習問題（P.98）

（1）プロセススイッチを頻繁に行うと何が問題かを考えその対策を考えよ．

［解答］

プロセススイッチを行うにはOSオーバヘッド（管理コスト）を必要とするのでCPUを消費する．またプロセスがCPUを短時間しか使用しないにもかかわらずプロセススイッチがプロセスの意に反して行われる状況はOSオーバヘッドによるコンピュータシステムの性能劣化と言わざるをえない．

（2）図4･11のサンプルプログラムを動かしてみよ．親プロセスは最後に子プロセスの終了コードを表示しているが図4･9の上位バイトの数値ではない．これを修正して分かりやすい表示に改良せよ．＜付録D＞

［解答］

付録D 参照

（3）プロセス生成の具体的な方法としてUNIX系OSにはforkがある（4章4-2-1）．子プロセスは親プロセスと全く同一のクローンであるがその利点と欠点について考察せよ．

［解答］

OSは親プロセス領域を同一サイズのメモリを確保してそこにコピーするだけで済むのでインプリメンテーションは容易となる．しかし子プロセスが親プロセスと同一のメモリ領域を使用するか不明であり，初期のUNIXではメモリが無駄になる可能性があった．

（※補足説明）

UNIXはこの弱点がcopy on writeという方法で改善されている．この方法は，forkで子プロセスが生成されると，親子プロセスの全領域がwrite禁止とされる．親子のプロセスがそれぞれfork後の命令を実行し，ストア系の命令がデータ部に実行されると書込み禁止の割込みが生じる．子プロセスの場合はこの時点でcopy on writeモードにあるページであることをカーネルは判定し新たなページ

を確保し，そこに親のページ内容を始めてコピーする．この方法でwriteした部分だけがコピーされるので，read only の命令部分（テキスト部）ならびに未参照部はコピーされることがなくメモリの節約につながる．

（4）日付・時間を表示するプログラムを作成せよ．ただし日本時間とすること．
＜付録E＞

［解答］

付録E 参照

（5）一つの仕事を分担し並列処理する方法は情報システム構築に有効である．図4･3と図4･4のようにタスクもしくはプロセスがアドレス空間を共有するスレッド方式と独立したプロセスファミリー方式がある．そこでそれぞれの利点と欠点について現在のCPUの仕組みなども考慮して論ぜよ．

［解答］

現在のCPUは主記憶を直接参照するのでなくキャッシュメモリをCPUチップ内にもちアクセスしている．また仮想記憶を実現しているため，DAT高速化とアドレス変換高速化のためにTLBを保有している．更に命令の先読みやパイプラインによる実行の高速化を図っている．これらの動作はプロセススイッチが起きると制御が乱れることで急ブレーキがかかるのと同じでCPUの速度は落ちてしまう．特にアドレス空間がプロセススイッチにより切り替わることで命令パイプラインだけでなく，TLBやキャッシュメモリの無効化が起き低速化の要因となる．

5章　5-5 演習問題（P.124）

（1）生産者と消費者問題（5章5-2-5）では排他制御すべき変数はどれか考察せよ．

［解答］

複数のプロセスで共通的に参照する変数のすべて．

(2) ログイン直後にファイルをオープンするとファイル記述子が3であることを確認するプログラムを作成せよ．＜付録F＞

[解答]

付録F 参照

(3) 親のプロセスが端末から入力メッセージ（例えば英単語）を受取り，その情報を子に転送する．子はその英単語を端末に表示し，それに対する日本語を入力し親に伝える対話プログラム（人間・英和辞典のような）を作成せよ．親子でパイプを通して相互通信するので双方向パイプである（図5･14)．
（※注意として親子の入出力が分かるように工夫すること）＜付録G＞

(a) fork()で子プロセスを生成する基本プログラムを作る（図4･11)．
(b) 親プロセスと子プロセス間の通信用パイプ2本を作る（rpfd, wpfd)．
(c) 親プロセスは入力用パイプrpfd[1],wpfd[0]を使用しないのでcloseする．
(d) 端末からstdinから文字を読み込む．（空文字入力で処理の完了とする）
(e) 端末入力文字を子プロセスに送る（wpfd[1]にてwriteを使用)．
(f) 翻訳結果を得るためにrpfd[0]にてreadで待つ．
(g) 子から翻訳結果を得たら端末に表示し，次の単語入力のため(d)へ．
(h) 子プロセスは使用しないrpfd[0], wpfd[1]をcloseする．
(i) 親プロセスから英単語を受け取るためwpfd[0]でreadする．
(j) readの内容を端末出力し，日本語に翻訳して入力する．
(k) 日本語翻訳結果を親プロセスにrpfd[1]を使いwriteする．
(l) (i)へ．
(m) 終了方法はどのようにするか考えてプログラムする．

[解答]

付録G 参照

(4) 本章5-3ではパイプの使用例として(who | grep yasu) を説明した．そこでこの処理を実行するプログラムを作成せよ．＜付録H＞

(a) 実行プログラム名をpipe5とするときプログラムの実行は: $./pipe5 yasu とする．

(b) argc, argv[ ] によりパラメータ（上記の yasu ）が入力されている確認する.
(c) pipe(pdf)を実行する．システムコール実行後は正常終了を確認する.
(d) whoを実行するプロセスを生成する．whoは使用しないpdf[0]をcloseする.
(e) 次にstdoutにpdf[1]をコピーするのでcloseする．そしてdup(pdf[1])を実行する.
(f) 準備ができたのでwhoを実行する(execlp(“who”,”” ,char(*)NULL)).
(g) 親プロセスはgrepを実行する子プロセスを生成する.
(h) grepのプロセスを実行するにあたり使用しないpdf[1]をcloseする.
(i) 次にstdinをpdf[0]にコピーするためにcloseしその後dup(pdf[0])を実行する.
(j) grep実行の準備をしたのでgrepを実行する．execlp(“grep”,”” ,・・・).
(k) このときwhoの出力を検索するためにargv[1]をパラメータに指定する.
(l) 親プロセスはgrepのプロセスを生成したら二つの子プロセスの完了を待つためにwaitを実行する.
(m) 両子プロセスが完了するのを確認し処理を完了する．このとき各々の子プロセスが使用したpfd[0],pfd[1]をcloseしておく.

[解答]

付録H 参照

6章 6-5 演習問題（P.147）

(1) 動的分割方式（6章6-2-2）ではジョブが完了した後に空き領域ができる．これらの未使用領域を管理する以下の方法がある．各々について考察せよ.
(a) 空きができる順に並べて管理し，そのサイズは無視する．パーティション作成は空きの先頭からサイズの条件を満たした未使用領域を割当て余分な領域は先頭の空きパーティションとする.

[解答]

空きパーティションの管理は容易である.

(b) 空きの領域はサイズの小さな順に並べて管理する．パーティション作成は上記(a)と同じであるが，余分な領域は再びサイズの小さな順に並べて管理する．

［解答］

ソートするオーバヘッドがある．新パーティション作成時には最も小さな空き部分ができてしまう．無駄が最小という特徴がありベストヒット法と呼ばれるが，果たしてシステム稼働全体としてベストだろうか?

(c) 上記(b)とは逆に空きの領域はサイズの大きな順に並べて管理する．パーティションの作成は(a)と同じであるが，余分な領域はサイズの大きい順に並べて管理する．

［解答］

サイズが大きい順に空きスペースがあるので，最初の空き領域が新パーティション作成時に使用できるので，空き領域のサーチは最短となる．しかし，それにより生まれる空きスペースの管理にはソートを改めて行う必要がある．この方法はワーストヒット法という．全体的によい方法かは不明である．

（2）動的分割方式におけるフラグメンテーション問題を解決するコンパクション方式がある．ジョブのパーティションを移動して空き領域をまとまる方法であるが，なぜこの方法は実現が難しいのか考えよ．

［解答］

フラグメンテーションを解消する手段として実行中のジョブ領域を一箇所に移動し空き領域を寄せ集める方法を説明した．しかし，ジョブ領域内にはアドレスを格納した領域（例えばポインタなど）が存在する．このためジョブ領域を移動するとこれらの内容が不正確になってしまう．この問題を解決するにはジョブ領域内に存在するアドレス格納部を判定する必要がある．つまり，各々のデータ属性を管理する情報が必要となる．このようなデータ管理可能なマシンはコンピュータ界古参のBurroughs社が1960年代前半に開発した先進的なマシンB5000がある．このマシンは各データの属性を示すタグを備えデータタイピングを可能にしていた．しかしこのマシンは野心的な試みで当時は例外的な仕様であり主流に

なり得なかった．また，OSがコンパクションを行うタイミングの問題，そのオーバヘッドと効果の関係など難しい問題もあった．

（3）ページ化されたメモリが一般的になっているがページサイズを決定する要因は何か考えよ．ヒントとして4KB/ページを基準としたとき1KB/ページと16KB/ページとした場合，何がどのように変わり，その利点と欠点を考えてみるとよいだろう．

［解答］

小さなページにすると無駄にする主メモリが少なくて済む．逆も真である．小さなページの場合はアドレス変換テーブルが大きくなるためメモリのオーバヘッド（仮想記憶を管理するコスト）が大きい．

（4）ページサイズが4,096バイトで論理アドレスが35,594番地のとき，論理ページ番号はいくつか．またページ内オフセット値はいくつか．

［解答］

[論理ページ番号] = [35594/4096] = 8

[オフセット値] = 35594 − 8*4096 = 2826

（5）複数のユーザが1台のコンピュータを対話モードで共同利用しているが主記憶が不足しておりメモリがボトルネック（スラッシング状態）になっている．仮想記憶をサポートしているこのOSは現在利用中の全ユーザのプログラムやデータ領域を主メモリ内に保持しているが，メモリネックのため多重度を下げる手段を考えている．このときどのユーザ領域をスワップアウトすればよいのか考えよ．

［解答］

近い将来メモリを使用することがないプロセスをスワップアウトする方法が考えられる．対話モードで利用しているユーザがいるならば，端末から文字入力中のプロセス空間をスワップアウトの候補とするのが妥当．スラッシング状態であるならば，ワーキングセットの大きなプロセスをスワップアウトの対象とすることで実メモリに空きができ問題を解決できる．

（6）プリページングを行うのに適したプログラムにはどのようなものがあるか.

［解答］

リアルタイム性が要求されるような処理はページフォールトによる処理の遅延は障害になる．例えば，ストリーミングサービスなどの音声，動画像再生などはこの類のプログラムといえる．また,リアルタイム性を要求されるような処理や,時間的な制約を伴う装置の制御を行う場合にも有効である．これらの処理ではプリページング以外にプロセス空間を実記憶にロック（固定）しページングやスワップアウトを行わない方法が場合によっては望ましい.

7章　7-5 演習問題（P.170）

（1）LRUアルゴリズムが有効なメモリ参照と無効なメモリ参照はいかなる場合か説明せよ.

［解答］

メモリ領域を一方向に参照するようなデータ領域はLRUに向いている．例えば順編成ファイルの読込み時に複数のバッファ領域を保有している場合など．もし，順編成用のバッファ領域であることが自明ならば，参照されたバッファ領域は直ちに破棄すべきである．OSが仮想ページの内容を破棄できる機能をもっている場合にこの方式が有効である．一方，大きなメモリ領域をアトランダムに参照するようなケースではLRUは必ずしも有効ではない．ただし，アトランダムという意味をより詳しく分析する必要があるだろう．真の意味でのアトランダムな参照はまれである.

（2）図7･11に示した4KBごとに実ページを管理するテーブル（RPCT）が32バイトとすると1GBの実記憶管理に必要なメモリはいくらか．このことから大容量の実記憶の場合の仮想記憶を管理するメモリオーバヘッド（管理コスト）が増大する問題について考察せよ.

［解答］

4KB/pageの場合1GBは

$1〔GB〕/4〔KB〕= 2^{20}〔KB〕/4〔KB〕= 2^{18}$ pages

である．したがって管理テーブル（RPCT）が32B（$= 2^5$B）の場合は，

$2^{18} \times 2^5 = 2^{23}$ B（= 8MB）

を必要とする．実記憶サイズが巨大化してもメモリの管理に必要とされるRPCTの量は相対的には同一である．しかし利用可能な実記憶量が16GBのようになってくると，4KB/pageのような小さなページサイズの場合，RPCTの量は128MBにも及び無視できる量ではない．このことから，利用可能な実記憶容量が増大した時点では小さなページサイズは見直しを行い，より大きなページサイズが望まれる．64ビットアドレッシングのアーキテクチャでは2MB, 1GBをページ単位とするマシンもある．

（3）プログラムはメモリを参照することで処理を進めるが，命令ならびにデータのメモリ参照に特徴がありそうなプログラムの例を想像してみよ．

［解答］

巨大なマトリックス計算では行・列の参照順序を調べ，データを行と列のどちらに格納すべきか検討する必要がある．また一見するとアトランダムな参照に思えるデータ領域であっても，参照の規則性や一部のホットスポット参照があるかもしれないので，注意深く観察してみる必要がある．

（4）ワーキングセットのサイズが小さいと思われるプログラム，逆に大きくなりそうなプログラムの例を考えてみよ．

［解答］

小さなループで実行が行われデータ領域は小さいケース．データ値によって処理するプログラムが変わり，かつ，データが大きな領域にばらまかれており，参照するデータの順序に規則性がないケース．

8章　8-3 演習問題（P.184）

（1）個人利用のPCにおいて仮想計算機を利用する必要性はあるか．

［解答］

組織によっては取り扱う書類などを限定し共通の形式を操作するソフトを使用するためにOSが限定されることがある．一方，プログラムを作成する場合などではプログラミング環境に合ったOSが必要になることがある．このような場合，複数のPCを用意するのは経済的にも場所やPC管理などの面を考えると仮想計算機の導入が望ましい．

（2）ファイルシステムはカーネルの動作モードと同じである必要があるか．カーネル化した場合の利点，欠点などを考察せよ．

［解答］

ファイルシステムは論理的な処理が大半であり実際の記録媒体への書込みと読込みの操作はカーネルに委ねる必要があるが，それ以外の処理はアプリケーションプログラムに近い性格である．アプリケーションプログラムとすれば機能拡張や処理法の変更，デバッグなどが容易に行え，かつソフトウェアの更新など迅速に対応可能となる．カーネルで実行する利点は割込みによるオーバヘッド削減による性能向上であろう．

（3）クラウドコンピューティングが広く利用されているが，仮想計算機の利用は有効か否か考えよ．

［解答］

ソフトのバグ対策，信頼性の向上，VMマイグレーションの使用によるマシン使用効率の向上，などがあり全体的な経済的利点がある．

9章　9-6 演習問題（P.223）

（1）イーサネットのMTUは1500オクテットである．このとき最長のペイロードはいくつか．

［解答］

IP, TCPヘッダの最小サイズ（各々20オクテット）から計算可能．

（2）DHCPサーバをネットワーク内に二つ以上を立ち上げたとき，クライアントに貸し出すIPアドレスの重複を避ける有効な方法を考えよ．

［解答］

各DHCPサーバが配布するプライベートIPアドレスをそれぞれのDHCPサーバにあらかじめ分けておくことで同一IPアドレスの割当て競合問題が起きないようにする．

（3）IPv4ではIPアドレスが32ビットである．192.168/21の表記は192.168.0.0から先頭の21ビットのサブネットマスクによるアドレス空間を示しているがいくつのIPアドレス数が利用できるのか．また，各々のネットワークではいくつのホストまで接続可能か．

［解答］

ネットワークアドレス，ホストアドレスの全ビットがそれぞれ0，1の例外を考慮すること．

（4）OSI基本参照モデルでは7層のレイヤーを定めている．したがって第3層以上のソフトウェア実現においても階層化した実装が望ましいか否かを多角的に考察せよ．このとき，図9・5のような構成を参考にしてみるのも一つのヒントになると思われる．

［解答］

2章演習問題（2）を参考にせよ．

（5）TCPとUDPでは用途が異なることを説明した．IP層のようにコネクションレスの信頼性が低いパケット通信により，電話を実現すると品質の低下が起こりそうに思えるが，それにもかかわらずなぜIP電話はVoIPとして実用化されているのか考察せよ．

［解答］

回線品質の向上によりパケットロスは低下傾向にあること，また電話は人間による判断があるので音声の途切れやノイズの発生があれば相互に問合せをし，意志の疎通が可能であり多少の品質劣化は実用上問題がない．品質とコストとのバ

ランスから使用者は選択が可能である．

（6）TCPヘッダ内のポート番号は16ビットである．表9・3に示されているようにクラアントに割り付けられるポート番号は動的，プライベートの範囲である．これらを前提に考えて，プライベートIP使用のネットワーク内のホストAとBが（図9･19のような）外部の同一Webサーバをアクセスしたとき全く問題は生じないのだろうか考察せよ．

［解答］

16ビットであるため65536のポート番号が論理的な範囲となる．クライアントに割り付けつけられるポート番号は表9･3のように動的，プライベートポート（49152-65535）であるので，実際には16384となる．したがって問題が全く生じないということはない．

（7）各種の機器によりネットワークは構成されている．図9・4に示されるOSI基本参照モデルの物理層，データリンク層，ネットワーク層に各々特化した機器名と機能について調べよ．

［解答］

リピーター：第1層である物理層でネットワークを延長する機器である．減衰などで電気的に崩れた波形を復元し次のネットワークに信号を伝える．機器によっては通信媒体である光ファイバからの信号を同軸ケーブルの信号に変換する．

ブリッジ：第2層であるデータリンク層でネットワーク間に流れるフレームを蓄積しチェックを行い接続された別のネットワークに正常なフレームを送出する．蓄積交換するため伝送速度の異なる通信媒体間の接続も可能にしている．各ネットワーク内のMACアドレスを記憶し次のネットワークにフレームの転送が必要か否かを判断できるよう高度化したラーニングブリッジもある．L2スイッチと呼ぶ場合もある．イーサネットのスイッチングハブはこの1種である．

ルータ：第3層であるネットワーク層の処理をする．パケット交換が主たる機能でありネットワークセグメント内へのパケットの送受信を行う．詳細は本文で述べられている．ソフトウェアにより高度な機能を各種備えることができ多種多様な機器が存在する．IPsec, VPN（Virtual Private Network）などセキュリテ

ィ機能を備えている機器もある．

ゲートウェイ：ネットワークへの玄関となるノードを呼ぶ場合もあるが異なるプロトコルをもつネットワークへの接続をする機能をゲートウェイと呼ぶこともある．この場合はOSI基本参照モデルの第4層以上のすべてのプロトコル変換を可能にする．携帯電話のメールをインターネットのメールに相互変換などが典型的な例である．機器やソフトのプロバイダによりゲートウェイの定義も異なるので注意.

10章 10-3 演習問題（P.242）

（1）付録Iを参考にTCPによるクライアント・サーバモデルのプログラムを理解する．

(a) 1台のコンピュータで基本的な動作の確認をする．このためコンパイルは（Ex. $cc -o server server.c）のようにしてコンパイル結果のファイル名をserverとする．

※（注意）サーバプログラムをコンパイルしバックグラウンドでサーバを動かすこともできるがここではプログラム内でstdout/stdinを使っているので使用できない．バックグラウンドでも使用できる応用に拡張するのも良い。Linux系ではバックグラウンドとして動作させるために，「&」をコマンドラインに入れる（Ex. $./server &）.

(b) 別のウインドウを開き，クライアントのプログラムを動かす．これでサーバの画面に動きがあるか否か確認する．＜付録J＞

(c) 二つ以上のクライアントを動作させて正確に動作することを確認する.

(d) サーバプログラムを拡張する．拡張はサーバのポート番号を外部から指定して設定する．agrc, argv[ ]を使う．プログラムでは int main(int argc, char* argv[ ]) とし，サーバ起動は，例えば（Ex. ./server 51200 &）のようにして動くようにする．ここで51200はポート番号である.

(e) 次にクライアントプログラムの拡張をする．クライアントはサーバのIPアドレスとポート番号を指定できるようにする．具体的には（Ex. $./client

127.0.0.1 51200）のように第1引数にサーバのIPアドレス，第2パラメータにサーバのポート番号を指定する．

(f) 上記の（d），（e）が動作することを確認する．

(g) 上記（f）が確認できたらサーバもしくはクライアントを別のコンピュータにインプリメントして2台以上のコンピュータで動作することを確認する．

［解答］

付録I, J 参照

(2) ドメイン名を外部からパラメータとして与えgethostbyname関数を使いhostent構造体を得て，その内容（オフィシャル名，ホストタイプ，アドレス長，エイリアス名一覧，IPアドレス一覧など）を端末に表示するプログラムを作れ．＜付録M＞

［解答］

付録M 参照