クラウドコンピューティングのための
OSとネットワークの基礎

工学博士　木下　宏揚 著

コ ロ ナ 社

まえがき

インターネットは電気，水道，ガスと同様に生活に欠かせないインフラとなっている。各家庭まで光ファイバにより高速なネットワークが整備され，スマートフォンなど移動体通信も進化し，シームレスでユビキタスな環境が実現している。高速なネットワークと仮想化技術の進歩によりクラウドコンピューティングが登場し，企業などのサーバの利用形態は大きく変化した。ユーザが接する計算機環境は，スマートフォンやタブレットなど多様化し，ストレージに関しても潤沢なネットワーク環境を活かしたオンラインストレージなど新しい利用形態が普及している。今後は IoT（Internet of Things）の普及によりあらゆるモノが接続され，ネットワーク経由での状態取得と操作が行えるようになる。一方課題としては，仮想現実感（Virtual Reality：VR）や拡張現実感（Augmented Reality：AR）など，より多くの帯域を必要とするアプリケーションへの対応やコンピュータウィルスやフィッシング詐欺，DDos 攻撃などセキュリティ問題への対応などが挙げられる。だれでも世界中に情報発信できるという点でインターネットは活版印刷に匹敵するブレークスルーであるが，歴史が浅いため法律面や倫理面ではこなれていない。今後は技術面だけではなく法律，倫理においても発展が望まれる。本書は，情報系に限定せず幅広い分野の学生を対象とし，インターネットのプロトコル TCP/IP の基礎とクラウドコンピューティングに必要なオペレーティングシステムや仮想化技術について解説する。また，一般のユーザが触れることのないデータセンターやインターネットサービスプロバイダで用いられている基盤技術についても触れている。

2017 年 6 月

木下　宏揚

目　　次

1. オペレーティングシステム

1.1 オペレーティングシステムの概要 …… 1
1.2 プ ロ セ ス …… 3
1.3 メ モ リ 管 理 …… 9
1.4 ファイルシステム …… 12
章 末 問 題 …… 20

2. インターネットの基礎

2.1 インターネットの歴史 …… 21
2.2 ネットワークの基礎 …… 22
2.3 OSI 参照モデルと TCP/IP …… 25
2.4 ネットワークに接続される装置 …… 29
2.5 ネットワークの単位 …… 31
章 末 問 題 …… 32

3. Ethernet

3.1 ネットワークアクセス層と Ethernet …… 33
3.2 フ レ ー ム …… 34
3.3 アドレス体系 …… 36

3.4 STP …… 36
3.5 無　線　LAN …… 38
章　末　問　題 …… 41

4. インターネットプロトコルの概要

4.1 インターネットプロトコル …… 42
4.2 IP アドレス …… 43
4.3 サブネット …… 45
4.4 ネットワークアドレスとブロードキャストアドレス …… 46
4.5 サブネットのネットワークの数と収容できるホスト数 …… 48
4.6 ドメイン名 …… 49
4.7 データグラム …… 51
4.8 プロトコル番号 …… 53
4.9 ICMP …… 54
4.10 ARP …… 55
4.11 IPv6 の特徴 …… 57
4.12 IPv6 のパケット …… 58
4.13 IPv6 のアドレス体系 …… 60
4.14 IPv6 アドレスの自動設定 …… 64
4.15 IPv4 アドレスの在庫枯渇と IPv6 への移行 …… 65
章　末　問　題 …… 66

5. UDP と TCP

5.1 コネクション …… 67
5.2 UDP …… 68
5.3 TCP …… 69

5.4 TCP のフロー制御 …… 71
5.5 TCP のコネクションの確立 …… 72
5.6 データの再送 …… 73
5.7 ポート番号 …… 73
章末問題 …… 74

6. DNS

6.1 IP 層のアドレス解決 …… 75
6.2 DNS によるアドレス解決 …… 76
6.3 ネームサーバの種類 …… 78
6.4 正引きと逆引き …… 79
6.5 リゾルバとドメイン名の検索 …… 81
6.6 DNS のセキュリティ …… 84
章末問題 …… 86

7. 経路制御

7.1 アドレスと経路制御 …… 87
7.2 経路のタイプ …… 88
7.3 経路制御テーブル …… 89
7.4 経路制御プロトコル …… 91
7.5 内部経路制御プロトコル …… 93
7.6 外部経路制御プロトコル …… 94
7.7 ISP と IX …… 95
7.8 ルータの冗長化 …… 96
7.9 サブネットと経路制御 …… 97
章末問題 …… 98

8. ネットワークセキュリティ

8.1 インターネットとセキュリティ …… 99
8.2 バッファオーバーフロー …… 100
8.3 ファイアウォール …… 102
8.4 パケットフィルタリング …… 103
8.5 プロキシサーバ …… 105
8.6 NAT …… 105
8.7 SSH …… 108
8.8 公開鍵基盤（PKI） …… 109
8.9 ディジタルフォレンジックスとセキュリティ監査 …… 114
章 末 問 題 …… 116

9. インターネットのアプリケーション

9.1 FTP …… 117
9.2 電 子 メ ー ル …… 118
9.3 WWW …… 129
9.4 NTP …… 134
9.5 ネットワークの診断 …… 136
9.6 VoIP …… 138
章 末 問 題 …… 139

10. 仮　想　化

10.1 仮 想 化 と は …… 140
10.2 コンピュータのハードウェア …… 142

10.3 SDC：計算機の仮想化 …… 146
10.4 SAN：ストレージの仮想化 …… 158
10.5 SDN：ネットワークの仮想化 …… 165
章 末 問 題 …… 176

11. クラウドコンピューティング

11.1 クラウドの定義 …… 177
11.2 サービスモデル …… 179
11.3 クラウドの配備モデル …… 180
11.4 オブジェクトストレージ …… 181
11.5 OpenStack …… 182
11.6 クラウドとセキュリティ …… 187
章 末 問 題 …… 187

索 引 …… 188

1 オペレーティングシステム

1.1 オペレーティングシステムの概要

家電に組み込まれたコンピュータからスマートフォン，クラウドコンピューティングまでオペレーティングシステム（Operating System：OS）は情報処理に欠かせないソフトウェアとなっており，以下のような機能を持っている。

1. ディスク装置やディスプレイなど機種ごとに異なるハードウェアの複雑な操作手順を抽象化あるいは仮想化し，システムコールとしてアプリケーションプログラムに提供する。
2. **マルチユーザ**（1 台のコンピュータを同時に複数のユーザが使用可能）と**マルチプロセス**（複数のソフトウェアを見かけ上同時に実行），**マルチスレッド**（multi thread，複数の CPU を持つ**マルチプロセッサ**（multi processor）環境における一つのプロセスの中で平行して同時に複数の部分を実行）を実現する。
3. 一つのハードウェアに複数のプロセスから同時にアクセスの要求が発生した場合にアクセスの順序などを計画する排他制御を行う。
4. 各プロセスにプロセス自体や処理の対象となるデータを格納するメモリの割当てを管理する。
5. ハードディスクなどのストレージを抽象化し，データやプログラムをファイルの形で格納するファイルシステムを提供する。
6. 各ユーザが実行するプロセスや所有するファイルをほかのユーザから保

護するアクセス制御を実現する。

7. 電源投入時からのシステムの起動と省電力制御やメモリ以外のほとんどの電源をカットするサスペンド（suspend），メモリの内容をディスクに待避して完全にシステムを停止するハイバーネーション（hibernation）などの電源の管理を行う。

プロセス（process）とはプログラムの実行の単位である。**マルチプロセス**（multi process）の機能は CPU を時分割（time sharing）で利用する，TSS（**タイムシェアリングシステム**）により実現されている。TSS はプログラムの実行時間を細かい時間区分に区切り，複数のプロセスを切り替えて実行させる方式である。切替えのスピードは 1 ms 程度と速いので，ユーザからはあたかも複数のプロセスが同時に実行されているように見える。

図 1.1 に BSD UNIX†を例にオペレーティングシステムの構成を示す。オペレーティングシステムの提供する中心的な機能をオペレーティングシステムの**カーネル**（核：kernel）という。カーネルのおもな機能には，プロセス制御，ファイルシステムの管理，通信機能を受け持つプロセス間通信であるソケットの管理，周辺機器などのデバイス制御の管理，システムコールインタフェースなどがある。カーネルは一般ユーザのアプリケーションプログラムとはハードウェア保護機構により分離され，専用のメモリ空間で動作する。この機能によりユーザのプログラムが暴走してもシステム全体に影響が出ないようにしている。**デーモン**（daemon）はカーネルではないが，オペレーティングシステムと

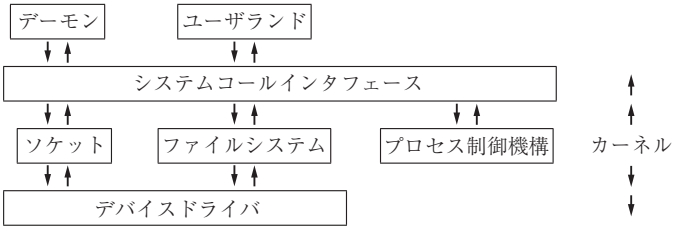

図 1.1　UNIX システムの構成

† M.K. Bostic, K. Karels, M.J. Quarterman and J.S. McKusick：The Design and Implementation of the 4.4BSD UNIX Operating System, Addison Wesley (1996)

密接に関連したさまざまな機能を提供するためのプログラム群である。デーモンはバックグラウンドで実行され，各種サービスの要求を監視し，要求があれば適切なサービスを行う。**ユーザランド**は，一般のアプリケーションプログラムなどのユーザの権限で動くプロセスである。**デバイスドライバ**はネットワークインタフェースやハードディスクなどのデバイスごとの違いを吸収し操作を抽象化する。プロセス制御機構はプロセス実行の制御とメモリ管理を行う。**スケジューラ**は，タイムシェアリングシステムにおける各プロセスへの CPU 割当てを優先度に応じて計画する機構である。**メモリ管理**は，各プロセスに対するメモリ割当てを行い，プロセス固有のメモリ空間をほかのプロセスから保護する。また，実際のメモリより大きなメモリ空間を提供する仮想記憶の管理を行う。

仮想記憶にはプロセス全体を待避するスワッピングシステム（swapping system）とメモリをページ単位で待避するページングシステム（paging system）がある。また，ネットワークと関連の深い**プロセス間通信**を担当する。プログラムは最初は実行可能ファイルの形で存在し，実行時にプロセスとなる。システムコールにより，プロセスの生成，終了，実行の同期などの制御が行われる。通常，プロセス（子プロセス）はプロセス（親プロセス）から生成される。例えばあるコマンドが実行される場合，そのプロセスはシェルを親プロセスとした子プロセスとなる。アプリケーションプログラムは，ハードウェア資源やメモリ割当ての要求，その他オペレーティングシステムの提供する機能はすべて**システムコールインタフェース**を介して行う。システムコールは C 言語や Perl などにより記述されたプログラムから利用することが可能である。

1.2 プ ロ セ ス

〔**1**〕 **プロセスとはなにか**　プログラムは**プロセス**と呼ばれる単位で実行される。**コンテキスト**（context）はプロセスに関するさまざまな情報を表し，アドレス空間の内容や環境を示す**ユーザレベル状態**とスケジューリングに関する変数や資源管理，識別子などを示す**カーネルレベル状態**がある。ユーザプロ

セスは**ユーザモード**と**カーネルモード**の 2 通りの動作を行う。ユーザモードはプロセスが動作する通常のモードであり，ハードウェア資源などにはアクセスできない。プロセスがシステムコールを呼び出すとカーネルモードに移行し，特権モードによりハードウェアなどにもアクセス可能になる。プロセスは**プロセス識別子**（Process Identifier：PID）により管理される。init はプロセスの初期化に関するプロセスで PID 1 が割り当てられる。すべてのプロセスの親プロセスをたどっていくと init が最初の親プロセスになっている。すなわち，プロセスは init を起点とするプロセス集合の階層構造を作る。

〔**2**〕 **プロセス管理**　　プロセスが使用するメモリ領域は 3 種類ある。

(1) **テキストセグメント**：テキストセグメントは，プログラムの機械語命令本体をロードする読出し専用の領域であり，ディスク上の実行可能ファイルより直接読み込み配置される。

(2) **データセグメント**：データセグメントは，プログラムで使用される変数データが格納される読書き可能な領域である。データセグメントには変数の初期値があらかじめ決定している**初期化データ部**と，初期値が代入されていない**非初期化データ部**がある。前者はテキストセグメントと同様にディスクよりロードされ，後者は一般に 0 で初期化される。データセグメントのサイズはプログラムの実行開始時に決定しているが，システムコールにより拡張・縮小が可能である。

(3) **スタックセグメント**：スタックセグメントは，データセグメントと同様に変数データが格納されるが，非初期化データのみ取り扱う。サイズはプログラム実行開始時には未定であり，実行後に必要に応じて自動的かつ動的に拡張される。スタックセグメントの内容は 0 で初期化されるが，これはプログラムの実行開始時のみなので，ユーザが初期化をつねに意識してコーディングする必要がある。

テキストセグメントとデータセグメントの初期化データ部はプログラムの実行開始時にすべてメモリにロードされている必要はない。また，すべて読み込むと実行時にその分時間がかかることになる。そこで実行につれ必要な部分だ

けがディスクよりロードされる。これを**デマンドページング**（demand paging）方式と呼ぶ。これらのメモリ領域とは別に，ユーザプロセスからはアクセスできないがカーネルによるプロセス制御に用いるユーザ領域がある。

〔**3**〕 **コンテクスト切替え**　一つのプロセッサでマルチプロセスを実現するための基本的な技法が**マルチプログラミング**である。マルチプログラミングは**タイムシェアリングシステム**により，CPU を時間的に分割して使用する。プロセスの切替えは**コンテクスト切替え**（context switching）により行う。

プロセスは生成と消滅およびコンテクスト切替えにより，プロセス生成時の中間状態からプロセス実行時の状態（走行可能状態，停止状態，事象待ち状態），プロセス消滅時の中間状態へと変化する（**図 1.2**）。コンテクストには，プロセスの実行に必要なテキストやデータが置かれるメモリ空間，プログラムの実行箇所を示すプログラムカウンタ，スタックの位置を示すスタックポインタ，プロセッサ内のレジスタのイメージなどが含まれる。システム全体の速度の向上のためには，コンテクスト切替えをいかに効率よく行うかが重要になる。コンテクスト切替えが発生するきっかけには 2 種類ある。一つは，ほかのプロセスからの通信待ちや入出力装置などの資源が利用可能になるまで待つなどそのプロセス以外の状態が変化するまで実行を続けることができない場合に，プロセスの側からコンテクストの切替え要求を出す自発的コンテクスト切替えがある。もう一つは，プロセスに割り当てられた時間だけ実行し終わったり，より高い優先度を持つプロセスが走行可能になったりした場合に，カーネルがコンテクストを切り替える強制的コンテクスト切替えがある。

〔**4**〕 **スケジューリング**　スケジューリングはプロセッサがいつどのプロ

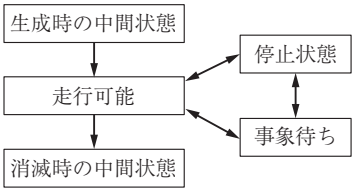

図 1.2　プロセスの状態

索　　引

【あ】

アドレス解決 *55, 75*
アプライアンスサーバ *140*
アプレット *133*

【い】

イーサネット *33*
一方向性ハッシュ関数 *109, 124*
インスタンス *183*
インターネット
　プロトコル *42*
インターネット VPN *170*

【う】

ウェルノウンポート *106*

【え】

エニーキャスト *60*
エフェラメルポート *106*
エンタープライズサーバ *140*
エントリ *89*

【お】

オクテット *34*
オープンリレー *127*
オンプレミス *180*

【か】

回線交換 *25*
仮想記憶 *9*
仮想マシン *146*
仮想マシンモニタ *146*
仮想 OS *146*
カーネル *2*
カーネルモード *4, 150*
間接経路 *89*
完全仮想化 *149*
完全性 *111*
慣用鍵暗号 *109*

【き】

逆引き *77, 79*
キャッシュサーバ *79*

【く】

クラウドコンピュー
　ティング *140*
クラスフルモデル *44*
クラスレスアドレス *45*
クラッカー *99*
グローバルスコープ *60*
グローバル IP アドレス *106*

【け】

経路制御 *27, 42*
経路制御表 *45, 54, 89*
経路制御プロトコル *91*
ゲートウェイ *30*

【こ】

広域 Ethernet *172*
公開鍵 *124*
公開鍵暗号 *109*
交　換 *25*
コネクション *27, 42, 69*
コネクション指向 *67, 104*
コネクションレス *42, 67, 104*
コリジョンドメイン *31*
コンテキスト *3*
コンテクスト切替え *5*
コンテナ *133, 154*
コンバージドシステム *165*

【さ】

サイト *32*
サブネット *45*
サーブレット *133*

【し】

資　源 *142*
システムコール
　インタフェース *3*
準仮想化 *149*
シリアル転送 *142*
シンクライアント *56*
シンボリックリンク *17*

【す】

スイッチ *30*
スイッチングハブ *30*
スケジューラ *3*
スケジューリング *5*
スタックセグメント *4*
スーパーネット *98*
スーパーバイザモード *150*
スーパーブロック *14*
スライディング
　ウィンドウ *71*
スラッシング *6*
スワッピング *6, 10, 11*
スワップ *10*

【せ】

静的経路制御 91
正引き 76, 79
セカンダリネームサーバ 78
セキュリティホール 99
セグメント 31
セッション鍵 124

【そ】

ソケット 28, 72, 106
ソースルーティング 52
ゾーンデータ 77

【た】

タイムシェアリング
　システム 2, 5

【ち】

直接経路 88

【て】

ディジタル署名 110
ディレクトリ 12
テキストセグメント 4
データグラム 42
データセグメント 4
テナント 183
デバイスドライバ 3
デマンドページング 5
デーモン 2

【と】

動的経路制御 91
特権モード 150
ドメイン名 49
トランジット 95

【に】

認　証 111
認証局 CA 111

【ね】

ネクストホップ 89
ネットマスク 45
ネットワーク 31
ネットワークアドレス
45, 46
ネットワークインタ
　フェース 27
ネームサーバ 76

【は】

ハイパーコンバージド
　システム 165
ハイパーテキスト 130
ハイパーバイザ型 147
ハイパーリンク 130
ハウジング 179
パケット 25
パケット交換 25
パケットフィルタリング 102
バ　ス 142
ハッカー 99
バッファオーバーフロー 100
ハードリンク 16
パラレル転送 142

【ひ】

ピアリング 95
ピアリングルータ 93
秘匿性 111
秘密鍵 124

【ふ】

ファイアウォール 102
ファイル 12
ファブリック 172, 183
フォワーダーズ 83
フォワーディング 30
物理マシン 146
プライベート
　IP アドレス 106
プライマリネームサーバ 78
ブラウザ 129
フラグメント 52
ブリッジ 30
フルルート 95
フレーム 34
プロキシサーバ 102
プロジェクト 183
フロー制御 71
プロセス 2, 3
プロセス間通信 3
プロセス識別子 4
ブロードキャスト 43, 60
ブロードキャスト
　アドレス 45, 47
ブロードキャスト
　ドメイン 31, 43, 55
プロトコル 25
プロビジョニング 183

【へ】

ベアメタル 184
ページング 6, 10
ヘッダ 28

【ほ】

ホスティング 180
ホスト 29
ホスト名 49
ホスト OS 型 147
ボーダールータ 93
ホップカウント 92
ポート 54
ポート番号 73, 104, 106
ポートフォワーディング 107

【ま】

マウント 12
マッシュアップサービス 133
マルチキャスト 43, 60
マルチスレッド 1
マルチプログラミング 5
マルチプロセス 1, 2
マルチプロセッサ 1

マルチユーザ 1

【み】

ミドルウェア 180

【め】

メインフレーム 140
メモリ管理 3

【ゆ】

ユーザモード 4, 150
ユーザランド 3
ユニキャスト 43, 60

【り】

リゾルバ 81
リピータ 31
リモートホスト 29
リンク 16
リンクカウント 15
リングプロテクション 151
リンクローカルスコープ 60

【る】

ルータ 30
ルーティングドメイン 92
ルートディレクトリ 13
ルートネームサーバ 78
ルールリスト 103

【れ】

レイヤ 2 スイッチ 31
レイヤ 3 スイッチ 30, 31
レイヤ 7 スイッチ 31

【ろ】

ローカルホスト 29
ロギング 102, 114
ロードバランサ 31

【A】

Ajax 133
ARP 56
AS 92, 94

【B】

Base64 122
BGP 92
bind 81

【C】

CGI 133
CIDR 96, 98
CISC 144
CSMA/CA 41
CSMA/CD 35, 40
CVSS 116

【D】

DANE 126
DNAT 107, 184
DNSBL 128
DNSSEC 85, 126
dynamic routing 91

【E】

EGP 92
Ethernet 33
Ethernet アドレス 36
EUI-64 61

【F】

forwarding table 89
FQDN 49, 81
FTP 117

【H】

HTML 129
HTML5 131
HTTP 129
HTTP/1.1 130
HTTP/2 130

【I】

IaaS 179
IANA 94, 106, 122
ICMP 54, 137
IGP 92
IMAP4 119, 121
inode 14
IP 42
IP アドレス 28, 43
IP-VPN 169, 172
ISATAP 65
ISP 23, 95
IX 96

【J】

Java 133
JavaBeans 134
JSON 132
JSP 134

【L】

LAN 22

【M】

MAC アドレス 36, 55
Maildir 119
make 134
MDA 119
MIME 122, 130
MMU 9
MRA 119
MSA 119
MTA 118
MTU 52
MUA 118

MX 120

【N】

NAS 161
NAT 105, 138
NIC 75
NIS 76
NTP 135

【O】

OpenStack 182
OP25B 124, 127
OSPF 92

【P】

PA アドレス 96
PaaS 179
Perl 133
PHP 133
PI アドレス 65
ping 137
PKI 110
POP3 119, 121

【Q】

QoS 58

【R】

RAID 18
RAM 144
RARP 56
REST 182, 186
RESTful 132
RFC 22
RIP 92
RISC 144
ROM 88, 145
routing protocol 91
routing table 89

【S】

SaaS 179
SAN 161
SASL 124
SMTP 118
SMTPAUTH 124
SMTP-STS 126
SNAT 107, 184
SSH 108
SSL 113
SSL/TLS 123
STARTTLS 123
static routing 91
syslogd 115

【T】

TCP 69
TCP セグメント 69
TCP/IP 26
Tier 95
TLS 112
traceroute 137
TTL 52, 54, 137

【U】

UDP 68
UDP データグラム 68
URL 130

【V】

VLAN 168
VLSM 97
VMM 146
VRF 168

【W】

WAN 22, 23
Web アプリケーション 121
Web メール 121
Web API 132
WebSocket 132
Web-of-trust 125
Web2.0 130
WWW 129

【X】

X.509 111
XML 132

【数字】

3 ウェイハンドシェーク 72
6to4 65

—— 著 者 略 歴 ——

1985年　電気通信大学電気通信学部電波通信学科卒業
1987年　東京工業大学大学院理工学研究科修士課程修了（電気・電子工学専攻）
1990年　東京工業大学大学院理工学研究科博士課程修了（電気・電子工学専攻）
　　工学博士
1990年　東京工業大学助手
1994年　玉川大学講師
1995年　神奈川大学助教授
2002年　神奈川大学教授
　　現在に至る

クラウドコンピューティングのための OS とネットワークの基礎
The Basis of Operating System and Networks for Cloud Computing

© Hirotsugu Kinoshita

2017 年 8 月 25 日　初版第 1 刷発行　★

検印省略

著　　者　木下宏揚（きのした ひろつぐ）
発 行 者　株式会社　コ ロ ナ 社
　　代 表 者　牛 来 真 也
印 刷 所　三 美 印 刷 株 式 会 社
製 本 所　有限会社　愛 千 製 本 所

112–0011　東京都文京区千石 4–46–10
発 行 所　株式会社　**コ ロ ナ 社**
CORONA PUBLISHING CO., LTD.
Tokyo Japan
振替 00140–8–14844・電話(03)3941–3131(代)
ホームページ　http://www.coronasha.co.jp

ISBN 978–4–339–02876–8　C3055　Printed in Japan　（新井）

JCOPY　＜出版者著作権管理機構 委託出版物＞
本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は，そのつど事前に，出版者著作権管理機構（電話 03-3513-6969，FAX 03-3513-6979，e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー，スキャン，デジタル化等の無断複製･転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は，いかなる場合も認めていません。
落丁・乱丁はお取替えいたします。