

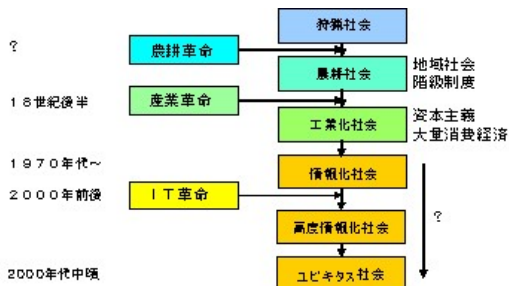
情報処理入門講義概要

西九州大学・非常勤講師：新井康平

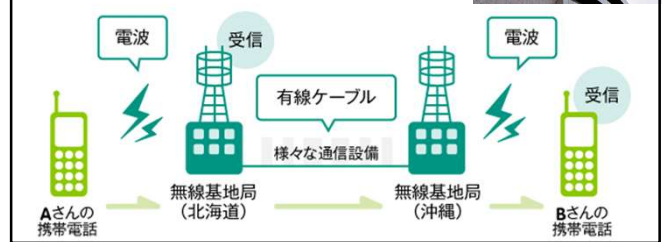
ITリテラシー

- コンピュータリテラシー**
 - ワープロソフトや表計算ソフトの利用、メール交換やWebページの閲覧などができる能力
- 情報リテラシー**
 - 組織内のデータベースやインターネットの情報などを分析することにより、業務に役に立つ情報に加工できる能力
- ビジネスリテラシー**
 - 業務改善や業務改革を実現するために、ITをどのように適用すればよいかを示すことができる能力。

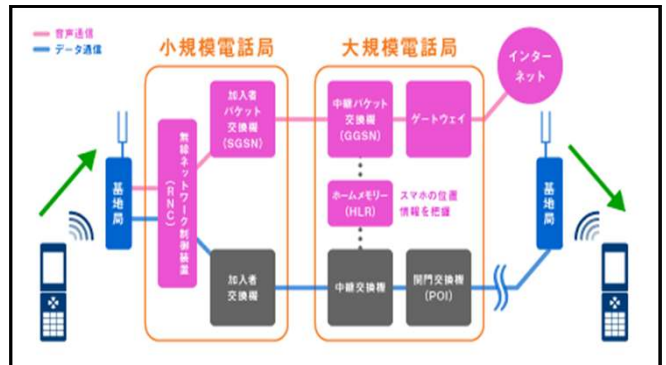
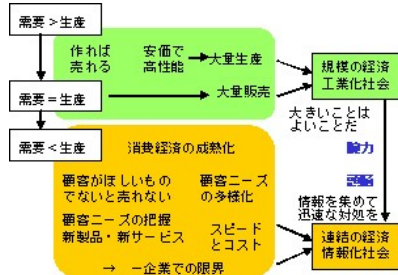
Industry4.0(AI,IoT、ビッグデータ)

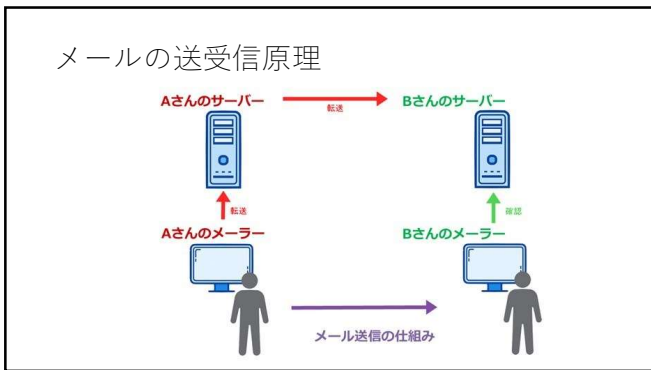
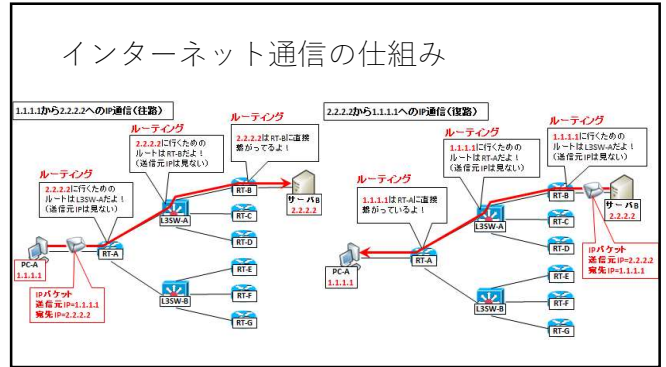
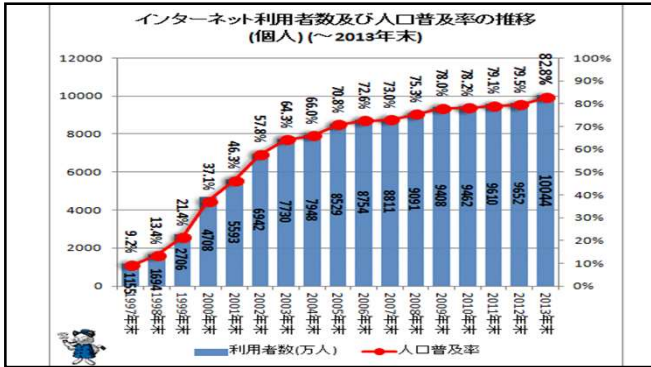


携帯電話の仕組み



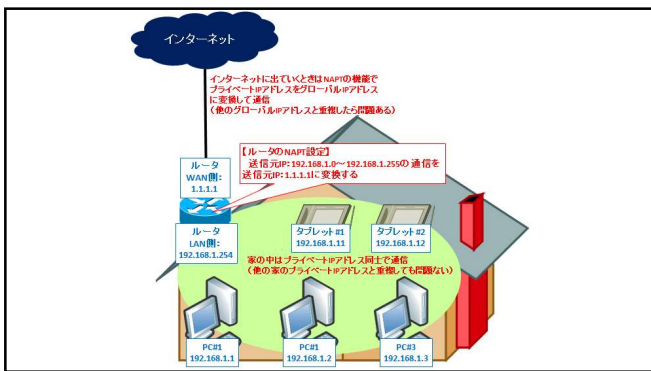
標準化→生産効率→個人仕様(リテール化)：顧客志向←ビッグデータ、AI学習





IPアドレスの確認

- Windowsキー + R
- ⇒ 『ファイル名を指定して実行』の入力欄に "cmd" と入力し、Enter
- ⇒ ポップアップした黒い画面 (MS-DOSプロンプト) に "ipconfig" と入力し、Enter



```

C:\WINDOWS\system32\cmd.exe
>>ipconfig

Windows IP 構成

イーサネット アダプター イーサネット:
    メディアの状態 . . . . . メディアは接続されていません
    接続固有の DNS サフィックス . . . . .
イーサネット アダプター イーサネット 3:
    接続固有の DNS サフィックス . . . . .
    リンクローカル IPv6 アドレス . . . . . fe80::1808:1068:7a8c:cfb2%14
    IPv4 アドレス . . . . . 192.168.179.6
    サブネット マスク . . . . . 255.255.255.0
    デフォルト ゲートウェイ . . . . . 192.168.179.1
    
```



授業計画（学習内容・キーワードのスケジュール）			
第1課	事前学習(予習)	今までのコンピュータの経験を振り返る	
	授業	学内システムについての説明（システムの利用方法）	4/9
	事後学習(復習)	講義中の操作について確認しておく	
第2課	事前学習(予習)	電子メールの基本機能を確認する	
	授業	学内システムについての説明（電子メール等の利用方法）	4/16
	事後学習(復習)	講義中の操作について確認しておく	
第3課	事前学習(予習)	コンピュータの動作原理について調べる	
	授業	コンピュータの基礎知識（コンピュータの動作原理等）	4/23
	事後学習(復習)	講義内容を振り返る	
第4課	事前学習(予習)	コンピュータの歴史について調べる	
	授業	コンピュータの基礎知識（コンピュータの歴史とこれから）	5/7
	事後学習(復習)	講義内容を振り返る	
第5課	事前学習(予習)	インターネットの動作原理について調べる	
	授業	インターネットの基礎知識（インターネットの動作原理）	5/14
	事後学習(復習)	講義内容を振り返る	
第6課	事前学習(予習)	インターネットの歴史について調べる	
	授業	インターネットの基礎知識（インターネットの歴史とこれから）	5/21
	事後学習(復習)	講義内容を振り返る	
第7課	事前学習(予習)	Windowsの操作、タスクインピット等について、操作の確認を行う	
	授業	Windowsの操作、タスクインピット等について、操作の確認を行う	5/28
	事後学習(復習)	教科書第1章を熟読しておく	
第8課	事前学習(予習)	Windowsの基礎およびタスクインピットと日本語入力	
	授業	Windowsの基礎およびタスクインピットと日本語入力	6/4
	事後学習(復習)	講義内容を振り返る	
	事前学習(予習)	教科書第2章を熟読しておく	
第9課	事前学習(予習)	MS-Wordの機能と操作（書体構成、日本語入力システム等）	
	授業	MS-Wordの機能と操作（書体構成、日本語入力システム等）	6/11
	事後学習(復習)	講義中の操作について確認しておく	

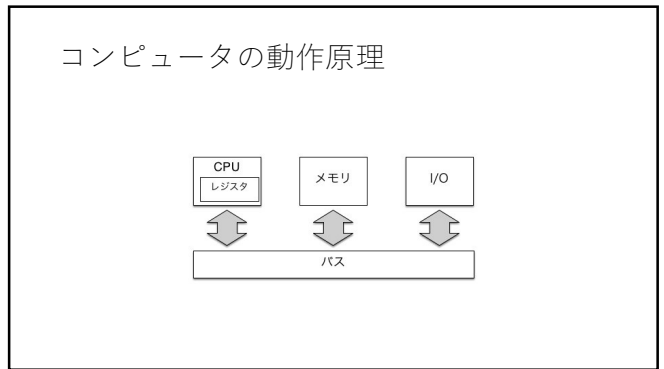
情報処理入門シラバス

科目名	情報処理入門		ナンバリング	開設科/年	神 埼
担当者	新井 謙平				
開 講 年 次	1	開講期	前期	単 位	1
開 講 学 科	社福・スポーツ		分類	共通教育科目 必修科目	
授業の概要	コンピュータを効率よく利用するための基礎知識を講義し、利用方法の演習を行います。具体的には、コンピュータの基礎知識、インターネットの基礎知識及び応用方法、文書や資料の作成方法の講義及び演習を行います。				
授業の到達目標	<ul style="list-style-type: none"> ①知識・理解 1) コンピュータやインターネットの仕組みを簡単に説明することができる。 ②思考・判断 1) 与えられた課題を視覚的に表現する際に、適切な表現方法を選択することができる。 ③関心・意欲 1) 文書を作成する際に、ソフトや文書の特徴、対象者等を考慮して創意工夫を行う。 ④技能・表現 1) 簡単なチラシや公文書を効率よく作成することができる。 				
学習方法	講義及びパソコンを用いた演習を行います。				
テキスト及び参考書籍	テキスト：30時間でマスターワード&エクセル2016、実教出版株式会社、参考書籍：コピキス社における情報基礎概論、学研出版				

第10課	事前学習(予習)	教科書第3章を熟読しておく	
	授業	MS-Wordによる文書の作成（文の入力、ページ設定、印刷、文書作成、複写・削除・移動等）	6/18
	事後学習(復習)	講義中の操作について確認しておく	
第11課	事前学習(予習)	教科書第4章を熟読しておく	
	授業	MS-Wordによる文書の作成（編集機能、箇条書き、文字飾り、表の作成、ビジュアルな文書等）	6/25
	事後学習(復習)	講義中の操作について確認しておく	
第12課	事前学習(予習)	今まで学んだ操作方法について振り返る	
	授業	MS-Wordによる文書の作成（簡単なチラシの作成演習）	7/2 7/9休講
	事後学習(復習)	講義中の操作について確認しておく	
第13課	事前学習(予習)	「公文書」について調べおく	
	授業	MS-Wordによる公文書の作成（公文書の構成と作成）	7/23
	事後学習(復習)	講義中の操作について確認しておく	
第14課	事前学習(予習)	文書の構成について確認する	
	授業	MS-Wordによる文書作成（演習）	7/30
	事後学習(復習)	講義内容を振り返る	
第15課	事前学習(予習)	文書の作成方法について確認する	
	授業	MS-Wordによる文書作成（演習）振り返り	8/8抽講
	事後学習(復習)	演習内容を振り返る	
第16課	事前学習(予習)	試験	8/6試験
	事後学習(復習)		
備 考	演習が中心の講義となりますので、精力欠乏しないようしてください。もし、欠席した場合は、各自で講義内容を把握するように努めてください。事前学習は特に必要ありません。講義終了後は、演習内容を確実に身につけるようしてください。なお、上記授業計画はあくまで予定であり、教員の判断によって若干の内容を変更する場合があります。		

情報処理入門・到達目標

比率	到達目標												合計					
	汎用能力要素						専門能力要素											
	態度・志向性		知識・理解		技能・表現		行動・技能・創造的応用力		態度・志向性		知識・理解			技能・表現		行動・技能・創造的応用力		
	1)	2)	1)	2)	1)	2)	1)	2)	1)	2)	1)	2)	1)	2)	1)	2)	1)	2)
評価基準・方法																		
定額試験																		
小テスト等																		
演習・授業外レポート																		
授業態度																		
受講者の発表																		
授業の参加度																		
その他																		
合計																		100



PC (Personal Computer=パソコン)のふたを開けてみましょう。その中には、様々な電子部品が詰め込まれています。黒い小さなポディーに数十本の銀色のピンが付いた「ムカデ」のような形をした電子部品が何個もあるはずですが、これらは、IC（アイシー、Integrated Circuit=集積回路）と呼ばれるものです。ICの内部には、数十個～数百万個の小さなトランジスタが入っていて、それらの組み合わせによって、コンピュータに必要な様々な機能が実現されます。膨大な数のトランジスタを持ち、高度な機能を実現するICを、特にLSI（エルエスアイ、Large Scale IC=大規模集積回路）やVLSI（バイエルエスアイ、Very Large Scale IC=超大規模集積回路）と呼ぶこともあります。

8本のピンでデータを入力するデジタルICがあったとしましょう。個々のピンの電圧を測定すれば、図1[拡大表示]のように0Vと5Vのいずれかになっています。0Vと5Vの羅列を数値の0と1に置き換えてください。01101010となりますね。これは8けたの2進数であり、10進数に変換すると106になります。Windowsに添付している「電卓」プログラムを使うと、2進数と10進数の変換ができるので、実際に試してみてください。

プロセッサとメモリーだけではPCが成り立ちません。なぜなら、PC本体にキーボードなどから情報を入力したり、ディスプレイなどに情報を出す必要があるからです。PC本体とキーボードやディスプレイなどの周辺装置を接続する機能を持ったICが、I/Oです。I/Oは、PC本体と周辺装置の電気的な特性の違いを調整します。I/Oは、周辺装置の種類ごとに専用のものが使われます。PC本体に装備されたコネクタの奥には、それぞれ専用のI/Oがあると考えてください。ネットワークカードやサウンドカードのような新しい周辺装置をPCに追加する場合には、I/Oを装備した拡張ボードをPCに装着することになります。

PCに装備されているメモリーは、バイト単位で情報を記録するのが一般的です。だからメモリー上の記憶容量を64MBから256MBのようにバイト(B=byteの略)で表すのです。メモリの中には、バイトに部屋番号が付けられているとされていて、それは1番地から999999番地まであります。メモリは、I/Oの中の部屋番号を指定して、その部屋番号はアドレスと呼びます。メモリーの授受は、アドレスを指定して、0番地から1番地まで、メモリーとI/Oに同じアドレスを振り分けられて、両者を区別できるように振振ってんでいます。

架空のICを使った架空のコンピュータの回路図(電気的な設計図)です。本物のコンピュータには、複数のメモリーとI/Oが装備されていますが、ここでは1つつずとしていきます。電源供給は省略し、データ授受、アドレス指定、動作制御のピンの接続方法だけを示しています。太い矢印は、複数のピンを表していますが、データ授受、アドレス指定、動作制御は、複数のピンで行われます。コンピュータの頭脳はプロセッサなので、皆さんがプロセッサになったつもりでコンピュータの動作を追ってみましょう。

これと同じことを8けたの2進数でやってみましょう。00000101 + 00000011 (10進数の5+3)を計算するならば、00000101 + (00000011の補数)とすればよいです。補数は、もとの数の0を1に、1を0に反転した数より1だけ大きい値になります。なぜなら、10進数の7と3の関係でも分かるように、補数は7を8から減らした1です。10進数の10と3の関係を7+3=10ですね。2進数の00000101と、それを反転した11111100を加えると、けた上りの1つ前の11111111になります。したがって、00000101とそれを反転した数より1つ大きい値(11111100+1=11111101)を加えれば、100000000の9けたとなります。8けたのサイズで計算を行うので、9けた目は無視されて00000000になります。すなわち00000011に計算すると、ちょうどけた上りの数は、11111101であり、これは-00000011をプラスで表した補数表現なので、そのままの補数を求める手順で「反転して1を足す」と覚えてください。

味覚	味覚コード
あまい	00000001
からい	00000010
しょっぱい	00000011
すっぱい	00000100
にがい	00000101

- 味覚のように、皆さんが自由に符号化できるデータであっても、それなりの工夫が必要です。「あまい」、「からい」、「しょっぱい」、「にがい」の5種類の味覚を、表1のように符号化したとしましょう。ここでは、8ビット(2進数で8けた)の数値を使っています。

- 単精度浮動小数点数形式と倍精度浮動小数点数形式の違いは、1つの小数点数を表すために使われるビット数です。単精度浮動小数点数形式は32ビットを使い、倍精度浮動小数点数形式は64ビットを使います。

味覚	味覚コード
あまい	00000001
からい	00000010
しょっぱい	00000100
すっぱい	00001000
にがい	00010000

あまい+からい=00000011

- 符号化ではどうでしょう？ここでは、個々の味覚ごとに1とするけたを変えています。これなら、どのように味覚コードを加算しても減算しても、上手く行きます。このように、符号化には、ちょっとした工夫が必要なのです。

- Windowsをお使いなら、プログラムは、拡張子が.exeまたは.dllとなったファイルとしてハード・ディスクに記録されています。ここでは、筆者オリジナルのダンプ・プログラムを使って、MyProg.exe(これも筆者が作ったオリジナルのプログラムです)の中身を調べてみると、[図1\[拡大表示\]](#)のようになっていることがわかりました。

下位4ビット	0	1	2	3	4	5	6	7
上位4ビット	空欄	空欄	空欄	空欄	空欄	空欄	空欄	空欄
1	ヘビィプログラム	設置制御1	1	1	1	1	1	1
2	ネットワーク接続	設置制御2	*	2	2	2	2	2
3	ネットワーク接続	設置制御3	*	2	2	2	2	2
4	空欄	設置制御4	*	4	4	4	4	4
5	問い合わせ	設置制御5	%	5	5	5	5	5
6	実行内容	実行制御	%	6	6	6	6	6
7	ヘルプ	設置制御6	*	2	2	2	2	2
8	検索	検索	(8	8	8	8	8
9	終了ボタン	終了制御)	9	9	9	9	9
A	空欄	実行制御	*	2	2	2	2	2
B	検索ボタン	検索	*	2	2	2	2	2
C	書込済み	ファイル制御	.	4	4	4	4	4
D	検索	グループ制御	.	4	4	4	4	4
E	シフトアウト	シフト制御	.	3	3	3	3	3
F	シフトイン	シフト制御	.	3	3	3	3	3

- ASCIIコードのコード体系です。7ビットの上位3けたと下位4けたから文字コードを得られるようになっています。例えば、大文字のAは、ASCIIコードで41となります。ここでは、16進数と呼ばれる数値の表し方が使われています。

リスト1●アセンブリ言語で記述されたプログラムの例

```

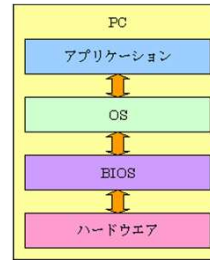
(動詞) (目的語) mov dword ptr [ebp - 4], 1 mov
dword ptr [ebp - 8], 2 mov eax, dword ptr [ebp - 4]
add eax, dword ptr [ebp - 8] mov dword ptr [ebp - 0Ch],
eax mov ecx, dword ptr [ebp - 0Ch] push ecx
push offset string "%d\n" call printf(00401080) add
esp, 8

```

- リスト1は、アセンブリ言語で記述されたプログラムの例です。mov (move)の略=データをコピーせよ)やadd (additionの略=加算せよ)などが、メモニックです。アセンブリ言語の文法は、実に単純です。「動詞+目的語」という文法で、ありませぬ。これは、英語の命令文と同じ構文です。動詞の部分には100000000という命令で、目的語の部分には「△△」というように命令の対象となるデータ(レジスタの名前、メモリのアドレス、1/0のアドレスなど)を指定します。目的語は、1つの場合も複数する場合もあります。目的語が複数の場合は、カンマで区切ります。ret (戻れ)のように、目的語のない命令もあります。リスト1のプログラムをマシン語に変換すると、[図1](#)のような数値の羅列になるわけです。

- リスト1に示したアセンブリ言語のプログラムをC言語という高水準言語で記述すると、リスト2のようになります。これは、1+2の加算結果を画面に表示するものです。高水準言語で記述されたプログラムは、コンパイラと呼ばれる変換プログラムによってマシン語に変換されてから実行されます。高水準言語で記述された1行のプログラムが、複数行のマシン語になります。すなわち、リスト2で使われているprintfのような高水準言語の命令は、複数のマシン語命令に変換されるのです。

```
リスト2●C言語で記述されたプログラムの例 #include
int main(int argc, char* argv[]) { a = 1; b = 2; c = a +
b; printf("%d\n", c); return 0; }
```

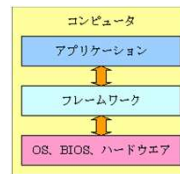


- アプリケーション、OS、BIOSという3種類のプログラム、およびハードウェアの関係を図3にまとめておきます。直接ハードウェアを制御していないため、アプリケーションの実行速度は遅くなりますが、プロセッサのスピードが驚くほど高速化した現在では、ほとんど問題にはなりません。

- BIOSのプログラムは、周辺装置を含めてコンピュータ全体を初期化します。PCを起動した直後の画面を見てください。BIOSが実行されていることを表す英語のメッセージが表示されることが分かります(図1)。BIOSが、コンピュータ全体の初期化を完了すると、ハードディスクからOSをRAMにロードします。この動作は「ブート・ストラップ」と俗称されます。ブート・ストラップ (boot strap) とは、ブーツを履くときに引っ張りあげる小さな皮紐のことです。小さなプログラムであるBIOS (皮紐) が、大きなプログラムであるOS (ブーツ) を引っ張り上げる (RAMにロードする) ことを表しています。この後のPCの動作は、OSにパトントンチされます。



- 「Java」と「.NET」といえます。Java (ジャバ) は米Sun Microsystemsが開発した技術で、.NET (ドット・ネット) は米Microsoftが開発した技術です。これらは「フレームワーク (framework)」と呼ばれます。フレームワークとは、OSとアプリケーションの間に位置し、コンピュータの機種の違いをすべて覆い隠してくれるプログラム群 (実行エンジンとクラス・ライブラリ) のことです。フレームワーク上で動作するアプリケーションは、コンパイル後に中間コード (特定のプロセッサに依存しないプログラム) に変換されます。フレームワークの持つ実行エンジンが、中間コードを解釈し、マシン語に変換して実行します。フレームワークは、GUI、インターネット、データベースなど様々な機能を実現するクラス・ライブラリ (プログラムの部品群) も提供します。プログラマは、クラス・ライブラリを利用してアプリケーションを作成し、高度な機能を容易に実現できます。



- Windowsのコントロールパネルから「システム」アイコンを起動し、デバイス・マネージャのプロパティを見れば、現在使用しているデバイス・ドライバを確認できます(図2[拡大表示])。周辺装置を追加または変更したときには、それに合わせてデバイス・ドライバも追加または変更する必要があります。

