

前回出た質問

Q. 【対面授業を実施する週（回）】 詳細は「学事
暦」（学生ポータルサイト掲載）を参照。

月水金の授業は 3,7,11 週（回目）を対面授業とする

火木土の授業は 4,8,12 週（回目）を対面授業とする

と大学からのメールに記載されていたのですが、対面授業の実施回はどうなりますか。

Q. 学事歴によると金曜日の授業は 3,7,11 週目が対面だと思われませんがこの授業はどうなりますか？また、事情により対面受講ができない場合はどうすれば良いですか？ Zoom で同時に配信していただくことなどはできますか？ その場合出席に認められないなどあるのでしょうか。

Q. 対面授業についてお伺いしたいのですが、対面授業は、スライドに記載がある、4,8,12 回目にそれぞれ

行うということによろしいですか？ その際はオンラインでは行わずに対面でのみということでしょうか？

A. 学事暦通り 1,7,11 週目に対面授業を行い、同時にその様子を配信します。また、プロジェクタで教室内でも公開します。

Q. この授業は必ず 3 回対面授業をする、という解釈
でよろしいですか？

A. はい、そうです。

Q. 3回の対面授業は、必ず登校すべきものなのではないでしょうか？それとも、引き続き自宅での受講が可能なものですか？

A. 私の理解では、「せっかく大学に入ったのに、登校できないでオンライン講義ばかりだと大学に入ったことによる差別化が不十分だから、少しでも登校させて差別化して欲しい」という文部科学省からの声に、大学が（いやいや）反応した結果であろうと思います。ですから正直全部オンラインでも困らないようにきちんと配慮しようと思います。

「試験だけオフラインで全員登校」ということは考えたのですが、前年度単位を落とした学生が試験を受けられないことになるので、諦めました。

Q. 対面授業はいかなかった場合の zoom などあるのでしょうか？対面授業についての詳細を伺いたいです。

A. 普通に zoom でも配信し、教室でも行おうと思っています。広い教室でマスクをしながらいつも通り zoom を配信し、それを登校してきた皆さんが大きなプロジェクタで見る、というシュールな光景が浮かびますが、止むをえませんね。

Q. 月水金にある授業の対面授業は 3, 7, 11 回目と大学からのメール (OchaMail) に書いてありました。対面授業該当日は zoom での授業は行わず、登校必須ということですか？

Q. 対面授業の日は必ず登校しないといけませんか。それともいつも通り Zoom での受講も可能なのでしょうか。

A. 「どうしても出席できない学生のために配慮せよ」とのことなので、いつも通り配信を行います。

Q. 質問なのですが、3回ある対面授業は完全に対面のみですか。それとも登校可能な人のみ登校し、オンラインでも配信していただけるのでしょうか？

Q. 登校しなければならない日が存在するのでしょうか？「登校しても良い」なのか、「登校しなくてはならない」のかを教えてください。大学から遠い所にいるので登校する場合は準備が必要です。

Q. 対面授業についてなのですが、必ず出席しなければならないのでしょうか？

Q 対面授業回の扱い (必ず対面で受講しなければな

らないのか、オンラインで受講してもいいのか等) について教えていただけると幸いです。

A. 「せっかくお茶大に入ったのだから、たまにはキャンパスにでていらっしゃい」というありがたい(おせっかいな?) 親心の現れだと思って、それに乗るのもよし、乗らないのも勝手だと思います。オンラインでの配信は通常通りで、出席も従来通り出席票のメールによる提出で行います。

Q. 対面授業は必ず学校に登校しなくてははいけませんか。

A. 私の講義ではそういうことはありません。特に不利にもなりません。ただ、他の先生がどのような予定かは不明です。

Q. 先日の Ochamail には、「月水金の授業は 3,7,11 回目の授業を対面授業とする」と記載されていたのですが、コンピュータアーキテクチャでは 4,8,12 回目を対面授業とするということでしょうか。

A. わたしのミスで 3,7,11 回目の授業というのが正しいです。

Q. 対面の授業はなるべく教室に行くことが望ましいのでしょうか？先生によってはあまりお勧めしないスタンスの先生もいらっしゃるのですが、3,7,11 週目に他の授業と都合を合わせながらできるだけ大学に行ったほうが良いのか、成績評価に加味されるのか知りたいです。よろしくお願ひ致します。

A. 対面授業を受けた、受けなくて成績をどうこうするつもりはありません。

コメント. 逆コンパイラについてのお話がとても興味深かったです。すごく便利だけど、自分で散々考えてかいたコードの可愛さを知っているので、あまり使いたくないなあとも思いました。

Q. OS なしで動くプログラムは作れるのか？という問いに対して、「動く」を広義にとらえればアセンブラを CPU に流し込むように書けば作れるが、何か目的をもって動くようなプログラムは OS なしでは作れないというように考えました。この考えはあっているのでしょうか？

A. 概ね合っています。では、現実にはこのようなプログラムは存在するのでしょうか？

まったく同一のハードウェア構成で、バージョンアップがあっても部分的な変更を許さず全体を更新する

もの。OSのように皆が使うものではないが必要とされる所には必須のもの。実は意外に身近にあります。パソコンのマザーボードのUEFIプログラム、いわゆるBIOSがそうです。

例え話になりますが、和裁が布を一枚のものとして扱い、代々洗い張りをして伝えてゆく、という思想なのに比べ、洋裁は服は人に合わせて作り、親子でもサイズが違えば別に作る（作らなくてはならない）。という思想です。

服の着心地は洋裁が上ですが、無駄な部分が多く出

ます。エコロジーの観点からは和裁が上です。
現在は洋裁の流儀でソフトウェア作成をするのが当然とされていますが、和裁の流儀を取り入れようという動きがあってもよいと思います。

Q. DVD-ROM に焼かれて綺麗にパッケージングされたのがアプリケーションソフトウェアなら Apple store などでインストールできるのはアプリケーションソフトウェアではないのですか？

A. まず、アプリケーションソフトウェア○、アプリケーションソフトウェア×です。このプリントの**昨年以前に出た質問**の項目を見て欲しいのですが、これらも立派にアプリケーションです。

私の考えですが、アプリケーションソフトウェアが成立するのは利用者が作者に対しアクセスする手段

がある（フィードバックが返せる）間だと思います。
アプリケーションソフトウェアが作者の死亡やサ
ポートの打ち切りでフィードバックが帰らなくなっ
た時に死ぬ、ということではないでしょうか。
もちろん作者が亡くなった場合でも有志がサポート
を引き継ぐことはあるわけです。

コメント. 2 ヶ月振りの授業、とても重たく感じてしまいました。感覚、すぐに戻せるよう努力します。

昨年度以前に出た質問

コメント. C コンパイラや emacs エディタとワードプロセッサやウェブブラウザはどれも「一般的なアプリケーション」ということですか！

補足. そうです。毎年これを聞いて驚く人がいます。歴史的には、エディタ、コンパイラなどをシステム寄りのプログラムとして、OS の一部として動かしたり、システム寄りのプログラムとして特別扱いした時代があります。現在では、OS(ないしその一部に

属するもの) という、純粹に一般のアプリケーションに奉仕するためのプログラムと、**アプリケーション** という、奉仕することを考えず、各々のプログラムが独自の動機に基づいて実行するという2種のものがあります。

Q. 質問のプリントの解答を教えてください。

A. すみません、もともとこのプリントは、ある程度授業を集中して聞いてほしいということで配ったものですので、授業中に答えを小出しにしていきます。まとめて答えてしまうと、一時覚えたような気になって、すぐに忘れてしまうのではないかと疑っています。

Q. X-Window と X-Windows の両方の表記が使われていましたが、これらは別物でしょうか？

A. おそらく私のミスですね。X-Window が正しいです。X-Windows というのは X-Window のアプリケーションが複数動いている状態を示しているのでしょう。なお、X-Window も一般のアプリケーションでしかなく、OS とは無関係です。但し、X-Window を作成するためにいくつかのデバイスが OS に追加されました。これが X-Window の本質であると強弁することは、できなくはないと思います。

Q. ソフトウェアというのは、アプリケーションプログラムの別名ですか？

A. ウィキペディアによると、

コンピュータプログラム（英：computer programs）とは、コンピュータに対する命令（処理）を記述したものである [1]。コンピュータが機能を実現するためには、CPU で実行するプログラムの命令が必要である。

また

ソフトウェア（英：software）は、コンピューター

分野でハードウェア（物理的な機械）と対比される用語で、何らかの処理を行うコンピュータ・プログラムや、更には関連する文書などを指す。ソフトウェアは、一般的にはワープロソフトなど特定の作業や業務を目的としたアプリケーションソフトウェア（応用ソフトウェア、アプリ）と、ハードウェアの管理や基本的な処理をアプリケーションソフトウェアやユーザーに提供するオペレーティングシステム (OS) などのシステムソフトウェアに分類される。

とあります。

要するにプログラムが一定数集まって文書が付き、
パッケージになったものがソフトウェアになります。

Q.OSの本質は見た目とは関係ない、という部分のイメージが沸かず理解できなかつたです。

A. 少し具体的な話をすると、UNIX(macOSX)のアプリケーションプログラムでは、プログラムの一番下に機械語があり、同じ層にOSのシステムコール呼び出しがあります(長尾先生のシステムプログラミング実習でやったと思います)。このシステムコールはアプリケーション単独ではできないことをするために呼び出されますが、このシステムコールの一覧にはプログラムを動かす場合の見た目を定義するも

のは一切含まれていません。そもそも見た目の変更などは OS がサービスしていないためです。

ではどこが担当しているか、というと、UNIX(macOSX)を立ち上げると自動で立ち上がるように設定されている X-Window(Darwin) というプログラム (一般のアプリケーションです) があり、それとやりとりをする形で見た目を整えています。

X-Window は OS そのものより大きく、その用途も多岐に渡るため、お茶大では X-Window そのものの講義はなく、伊藤先生の講義で X-Window に一皮

被せた OpenGL を扱う講義があります。なお、裸の X-Window を直接駆動するのはもはや流行らないようです。

私は OS そのものを担当しています。

Q. 最初の Mac はキーボードなしで～」の下の「メモリは 128KB」はこれで合っていますか？

A. 合っています。

Q. Linux と Windows、MacOS の違いは何ですか？

A. MacOS X と Linux はどちらも UNIX というファミリーに属する OS で、Windows は独自の OS です。UNIX とは 1969 年に産まれた OS で、出来た当時はキャラクターベースで動いており、標準となる入出力装置は ASR33 という 110 ビット/秒のテレタイプ

(物理的に印刷するタイプライタで、紙テープにも印字、紙テープから入力できた)でした。

Windows は PDP-11 の OS である RT-11 を元祖として (メーカーは違いますが)、初めてのマイコン 8080(8 ビット) 用の OS である CP/M(コンピュータ・モニタ; これも別メーカー) から 16 ビット用の 8086 用の OS である MS-DOS(ここからマイクロソフト)、16 ビット/32 ビット用の OS である MS-Windows と進化してきた OS です。MS-DOS までばモニタが付いたターミナル (テレタイプを高速化し

たものです) で動きましたが、MS-Windows からはマウス、キーボードが付き、640x480 ドット、16 色の VGA ディスプレイが必須になりました。

Q. 凍結されたプロセスは実行中だったときに割り当てられたメモリ空間を喰ったまま凍結されるのでしょうか (とっておかれてあるけれど、実際動いてないメモリ空間がプロセスを凍結することによって発生してしまうのでしょうか)

A. 答えはどちらもイエスです。しかし、OS 内部でスケジューラーが動くと、割り当てられたけれども触っていない記憶域がすぐに見つかり、メモリが足りなくなった時に真っ先に仮想記憶のメカニズムが働いてページアウトの対象になってディスクに書き

出され、メモリから消えてしまいますので心配する必要はありません。

Q. 他の授業では、Perl、Ruby、Python はインタプリタだと習ったのですが、本当は Java と同じようなコンパイラ・インタプリタなのですか？

A. そうです。現在使われている言語で純粋なインタプリタはまれな例外で、殆どはコンパイラ・インタプリタです。

Q. 複数のユーザが動かす複数のプログラムというのは1つのパソコン内での話ですか (A. そうです)、もしそうならいまいちそのような状況になる場面が想像つきません。

A. 例としてはあまりよくないかもしれませんが、皆さん方の知っている例だと、5 F の計算機室のマックはそうです。どれを使っても同じに動くように見えます。同時に2台を同じアカウントで使ったり、バックグラウンドでこっそり裏から入ったりできます。あれはそういうように管理されていて、

各マックにはあなた方全てがユーザー登録されています。正確には登録ファイルはサーバ上にあり、そのサーバを参照するようになっています。

Q. システムコールとは具体的に、fork() や execl() などとありましたがシステムコールとは「プログラムをプロセス化」つまり「プログラムを動いている状態にすること」ということですか？

A. それもありますが、もっと広く「他に方法がないからシステムコールがやる、ということ」すべてです。

Q. 携帯電話の例がよくわかりませんでした (OS の本質について)。もし理解していた方がよいならば、申し訳ございませんがもう 1 度説明してください。お願いします。

Q.OS なしで動くプログラムはあるか？のところで出た携帯電話についての例がよくわかりませんでした。できればもう一度聞きたいです。

A. 黒電話というのがありました。これは部品の組み合わせだけで出来ていて、制御一切を回線の向こう側 (NTT の交換局) に委ねており、OS の存在する余

地はありませんでした。（その後、回線のこちら側にある音や声、画像などを組み合わせて NTT 回線に接続し制御するものが産まれました。これは NAPLPS とよばれていました。三鷹市などにお住まいの方は覚えているかもしれないです。これは OS が動いていたと言えるかもしれません。）

さて未来の携帯電話があるとして、これはあるときは電話、あるときはモニタと自在に変形する、とします。これには OS の機能はあるでしょうか？当然あると思いますし、変形などは OS の機能になること

でしょうし、通話するのもまた OS が管理しているでしょう。

しかし通話するだけなら iPhone や現在のガラケー、もっといえばさらに過去の黒電話と互換性があり、通話が可能でしょう。ただ、未来の携帯だと OS を介してハッキングされてこっそり盗聴されても分からないかもしれません。iPhone も OS が通話を仲介しているのは同様なので、同じ危険があります。しかし、ガラケーだと OS の互換性がないため、盗聴しようがないかもしれません。黒電話は OS がないの

で、細工をしなければ盗聴不能です。

Android は Windows-PC 上でちゃんと動きますが、Windows-PC はタッチパネル方式ではないものが多いし、そもそも携帯電話の SIM が刺せないものが多いので、インターフェースを持っておらず、これらを使ったデバイスは動きません。

要は、OS がない初期には、デバイスは非力極まりないものが多い。OS があれば OS で頑張って機能を拡張し、カバーしようとする。しかしそれが常に良いほうばかりに使われるとは限らない。またそもそも

インターフェースが古く、どうやっても出来ないものは、最初から出来ない。ということです。

Q.OS の機能 1. コンピュータの機能を拡張する が理解しきれませんでした。もう少し説明していただけないでしょうか。

A. コンピュータで何かをする場合、山ほどプログラムを書いて、それをライブラリやシステムコールとして呼び出すのが普通になりました。そうしたライブラリやシステムコールがすでにあって、それを必要に応じて呼び出して使うというのが、OS の機能としてのコンピュータの機能拡張という意味です。

Q. 最近の家電（電子レンジや TV など）には OS が搭載されていますか？

A. はい。ITRON などを使っていると思います。

Q. されている場合、製品の種類ごとに別の OS がありますか？

A. いいえ、多分ないです。それより会社ごとに違った OS に手を出しているというほうがありそうです。

Q. プロセスについてももう少し詳しく知っていききたいです。

A. これから詳しい内容を説明していきます。

Q. コンパイラもプログラムの一つなのですか？

A. そうです。ごく普通のプログラムの1つに過ぎません。

Q. コンパイラは OS によって違うように思うのですが、コンパイラと OS には関係がないのですか？

A. コンパイラと OS は関係ありません。例えば Windows のもとで MacOS X のコードを出すためのクロスコンパイラは実在します。

Q. パソコンの OS とスマートフォンの OS は違う種類のものですか？

A. 種類は同じです。Windows PC で Android を動かすことができます。Windows はスマートフォンでは動きませんが。

コメント. プログラムの定義が「きょどう（挙動）を表したもの」とするならば、ありとあらゆることに対する手順をプログラムに見出すことができそう。

補足. はい、できると思います。

Q.OS の実体として Plan9 や Inferno は Windows の代わりに動くと黒板にありましたが、Mac や Linux の代わりににもなるのでしょうか？それとも特別 Windows だけの代わりをするのでしょうか？

A. もちろん、MacOS や Linux の代わりににもなります。