

ネットワーク システム管理 #08

たかさきこうや
1限 (09:00-10:45)

1

文字コード

- 問：03.txt、03-j.txt、03-e.txt、03-s.txtのファイルサイズについて、どうしてそうなるのかを推察し書きなさい
- 答：
元テキストは、全角30文字、半角9文字、改行3か所
- Shift_JISとEUCはどちらも2byteコード、且つ半角英数は1byteである
 - $30(\text{文字}) \times 2(\text{byte}) + 9(\text{文字}) \times 1(\text{byte}) + 3(\text{byte}) = 72\text{byte}$
- 元テキストは、UnicodeのうちUTF-8であるが、UTF-8では日本語は3byteになる
 - $30(\text{文字}) \times 3(\text{byte}) + 9(\text{文字}) \times 1(\text{byte}) + 3(\text{byte}) = 102\text{byte}$

3

UTF-8とUTF-16、BEとLE

- Unicodeは文字集合(どの文字を採用するかを決めている)
- UTF-8とUTF-16は、符号化方式(どの文字にどの番号を割り当てるかを決めている)
- UTF-16(2バイトコード)は、バイトの順番を変えられる
- Big Endian(ビッグエンディアン/BE)とLittle Endian(リトルエンディアン/LE)
- どちらが優れているというのではなく、処理系の向き不向き
- そしてそもそもmoreコマンドはUTF-16を処理できない

5

改行コード

- 問：03.txt、03_crlf.txt、03_cr.txtのファイルサイズについて、どうしてそうなるかを推察し書きなさい
- 答：
03.txtは改行コードが「LF」(UNIX改行)なので1バイト
03_crlf.txtは改行コードが「CR+LF」(Windows改行)なので2バイト
03_cr.txtは改行コードが「CR」(Mac改行)なので1バイト
- ファイルをUnicode+LFで作ると72バイト
Unicode+CRにしても72バイト
Unicode+CRLFにすると、改行コード3か所分データが増えるので75バイトになる

2

ISO-2022-JPだけ特殊

- ISO-2022-JPは、英数1byte日本語2byte
- ただし、全角半角の切り替え時にエスケープシーケンスが入る
 - エスケープシーケンスは3byte使う
 - 何byteなのかは場合により違う
- 切り替え箇所は、12か所ある
 - $30(\text{文字}) \times 2(\text{byte}) + 9(\text{文字}) \times 1(\text{byte}) + 3(\text{byte}) + 12(\text{箇所}) \times 3 = 108\text{byte}$

4

UTF-16の最小単位

- UTF-16においては、文字の最小単位は16bit(2byte)
- 半角英数であっても2byteを使用する
 - $30(\text{文字}) \times 2(\text{byte}) + 9(\text{文字}) \times 2(\text{byte}) + 3(\text{改行}) \times 2(\text{byte}) = 84\text{byte}$
 - 半角英数、改行の2byte目はnul文字
 - 従って、UTF-16 Little EndianもUTF-16 Big Endianも84byteになる

6

BOM (Byte Order Mark)

- Big EndianかLittle Endianかを区別するために、ファイルの先頭に『しるし』をつけるやり方がある
- これがついていると、処理系はBEかLEか、(UTF-8かUTF-16か)を見分けることができるようになる
- BOMは2byte(UTF-8の場合は3byte)で定義される
 - $30(\text{文字}) \times 2(\text{byte}) + 9(\text{文字}) \times 2(\text{byte}) + 3(\text{改行}) \times 2(\text{byte}) + 2(\text{byte/BOM}) = 86\text{byte}$
- BOM付のCSVファイル等を作ってしまうと、テキスト内に本来のデータではない文字が混ざるので良し悪しではある

7

UTF-16ファイルをどう読むか

- 元の文字コードがBEかLEかを処理系に対して教える
- 処理系が処理できるプログラムを使う
 - moreではなくlessコマンドを使うと、BOMさえついていれば正しく読める
- 処理系(more)に読める文字コードに、事前に変換してやる
 - nkfを使うなどする

8

先週のおさらい

- Emacsでファイルを編集できるようになる
- キーボードショートカットの概念を知る
- 楽をするための技術的な初期投資は怠らない

9

ショートカットの極意

- Windows系の一般的なアプリケーションだと、Alt-f Alt-s (Altキー + f Altキー + s)は、ほぼ「作業状態を保存する」操作にキーバインドされています
- 無意識にAlt-f Alt-sする癖をつけておけば、**「1時間かけて書いたレポートがWordが落ちてぶっ飛んだ」**みたいな悲劇を回避できる可能性があるので、手癖にしておくとも良いかもしれない

10

工数

- システム開発やネットワーク構築では、何か事象に取り組む際に「最長だとこれぐらいかかる」「最短でもこれぐらいかかる」と、作業見積もりを出す必然性が発生する
 - 「早くて4時間」「最長で3日」みたいな
- 「ある作業が終わるまでにどれぐらい時間がかかるか」を「**工数**」などという

11

工数の単位

- ある(特定の)人間が1日かければ終わる仕事は「**1人日**」(いちにんにち)という
 - ちなみに1人日は7-8時間程度で計算するのが普通
 - 外部に出す見積もりの場合移動コストも載せる
- ひとりが1ヶ月かけて終わる仕事は「**1人月**」(いちにんげつ)
 - 営業日計算なので、「1人月」の実稼働は20日程度
 - 土日もあるからね

12

人月単価

- 仕事早い人と遅い人が存在する
- 出来る仕事の種類が多い人と少ない人もいる
- 優秀なエンジニアの人月単価は高い
 - 僕が会社の帽子で仕事を請ける時は、稼働状況と仕事内容に拠りますが人日単価を5-10万で見積もります
 - 時給だと1-2万円
- そうでない人はそれなりに

13

単価のあげかた

- 仕事を早くする
 - Uber Eatsとかで早く配達する等の「速度が年齢に反比例する」ものは加齢とともに衰え価値が下がるからお勧めしない
 - 職人は「反復練習でミスを減らす」ことで仕事を早くする
- できる仕事の種類を増やす
 - ジェネラリストになる
- できる仕事の専門性を高める
 - スペシャリストになる

14

IT屋さんの生き方

- スペシャリストになる
 - あるアプリケーションや特定のシステムの専門家になるとそれだけで食っているが、土俵そのものが消滅すると突然食えなくなる
- ジェネラリストになる
 - フルスタックエンジニア：フロントエンド、バックエンド、ネットワークなどの全開発工程から運用までの面倒を見られる人(僕はこの人)
 - 上流工程：プログラマー→システムエンジニア→管理職
 - プログラマ35歳限界説

15

UNIXとemacsは

- 一旦置いて、ドメインの話再び
- <https://www.cuc.ac.jp/>にアクセスするたびに思う
- 「なんでこんなアドレスなんだろう？」
 - ※ドメイン≠所属

16

実はドメインは

- 世界中である程度の決まりごとがある
- **トップレベルドメイン**は...
 - 国別(地域別)ドメインである ccTLD(カントリーコード トップレベルドメイン)
 - gTLD (ジェネリックトップレベルドメイン) が複数 (.com/.net/.org/.info/.biz/.mobi 等...)
 - アメリカのみが使える特殊なドメイン (.edu / .gov / .mil)
 - その他

17

日本の場合

- .jp というドメインを使ってよいことになっている
- そして、その利用権限に関する裁量がJPRS という組織に委譲されている
 - (かつてはJPNICがその権限を持っていた)

18

更に、歴史的な経緯で

- JPRS (当時のJPNIC) は、「**学術組織用**」としてac.jpという**サブドメイン**をつくった
- ac.jpのうち、千葉商科大学にはcuc.ac.jpの利用権利を与えた
 - 慶應義塾大学にはkeio.ac.jp
 - 東京大学にはu-tokyo.ac.jp
 - 和洋女子大にはwayo.ac.jp

19

でも

- 世界中の何処からでも、<https://www.cuc.ac.jp/>と打てばCUCのWebサイトが見られるのは何故？
- 例えば、NYからでも、モスクワからでも、パリからでも、インターネットに繋がってさえいればCUCのページを見られるのはどうして？

20

推論

- 海外からでもCUCのサーバにアクセス出来るということは、その際にIPアドレスが分かる、ということ
- でも、どうしてwww.cuc.ac.jpという風にホスト名を打てば、www.cuc.ac.jpに相当するIPアドレスが分かるのだろうか？

21

例えば

- 自分のコンピュータには、そのホストとIPアドレスの対応表が既にインストールしてある？
- もしインストールしてあるとするなら、それはどのぐらいのファイルサイズになるか、予想してみよう

22

推論

- IPアドレスは約43億ある
- IPアドレスから、ホスト名が分からないといけない
- ホスト名は、更に多いはず
 - 同じIPアドレスに、複数のホスト名を付けられるから
- 一つのデータにつき、きっと100バイトぐらい使っちゃう

23

ということは

- $43億 \times 100\text{バイト} = 4300\text{億バイト}$
- 430ギガバイトの容量が必要！？
- でも、その辺に売ってるHDDの容量って、2000ギガバイト(2テラバイト)程度だよな…
- ありえない！

24

ていうか

- そのデータの変更をどうやって知るのさ！
- CUCが新しくホストやサーバを置きたくなったらどうするのさ！
- どこかに申請するの？

25

そこで

- 名前解決には、分散システムが取り入れられた
- 「DNS」という
- ドメイン・ネーム・システム

26

まず手元から

- 各ホストは、「自分が名前解決を行うために使う、DNSサーバの設定」をする
- Windows10だと
 - スタートボタン(右クリック)⇒ネットワーク接続⇒イーサネット⇒アダプターのオプションを変更する⇒イーサネット等(右クリック)⇒プロパティ⇒ネットワーク(タブ)⇒インターネットプロトコルバージョン4⇒プロパティ⇒全般
- に、DNSサーバ名がある

27

login.cuc.ac.jp の

- DNSサーバの設定は [cat /etc/resolv.conf](#) で確認できる

```

[kohya@hedgehog ~]$ cat /etc/resolv.conf
# Generated by NetworkManager
search chibagakuen.cc.cuc.ac.jp cc.cuc.ac.jp
nameserver 202.244.38.73
nameserver 202.244.38.74
[kohya@hedgehog ~]$

```

28

問合せ

- nslookup www.cuc.ac.jp とすると、202.244.38.73に

```

[kohya@hedgehog ~]$ nslookup www.cuc.ac.jp
Server:      202.244.38.73
Address:     202.244.38.73#53

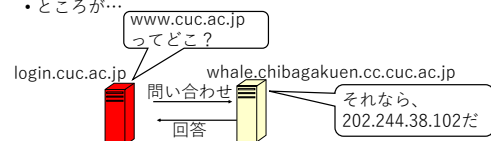
Non-authoritative answer:
www.cuc.ac.jp canonical name = bay.cc.cuc.ac.jp.
Name:   bay.cc.cuc.ac.jp
Address: 202.244.38.102

```

29

このDNSサーバの仕事

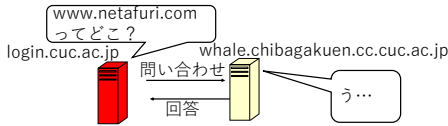
- 恐らく whale.chibagakuen.cc.cuc.ac.jp (202.244.38.73)
- これは、何も知らないCUCの各ホストから名前解決の問い合わせを受け、答える仕事をしている
- とところが...



30

whale.(後略) には

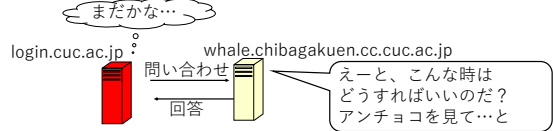
- 即答できる情報と、そうでない情報がある
- whale.(後略)は、CUC学内のことについては良く知っている
- でも、学外のことは良く知らない
- ましてや、www.netafuri.comというホストのIPアドレスなんて知るよしもない



31

だが、whale(後略)は

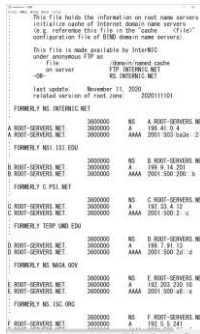
- アンチョコを持っていた!
- 「困ったときにはここに尋ねなさいリスト」
- ftp://rs.internic.net/domain/named.root
 - 最新のものはここ



32

これはなにか?

- ルートキャッシュファイル
- DNSの親玉のリスト
 - 「ルートネームサーバ」と言う
 - 超強力なサーバ



33

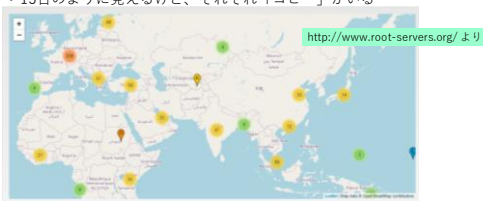
「ルートネームサーバ」は何処?

- 世界中に13台ある(ことになっている)
 - 米VeriSign社に2台(Aサーバ、Jサーバ)
 - ネットワーク管理団体ICANN(Lサーバ)
 - ヨーロッパのネットワーク管理団体RIPE-NCC (Kサーバ)
 - 米Cogent社(Cサーバ)
 - 米ISI (Information Sciences Institute)(Bサーバ)
 - 米ISC(Internet Software Consortium)(Fサーバ)
 - 米Maryland大学(Dサーバ)
 - 米NASA(Eサーバ)、米国防総省(Gサーバ)、米陸軍研究所(Hサーバ)
 - ノルウェーのNetnod(Iサーバ)
 - 日本のWIDEプロジェクト(Mサーバ)

34

「ルートネームサーバ」の位置

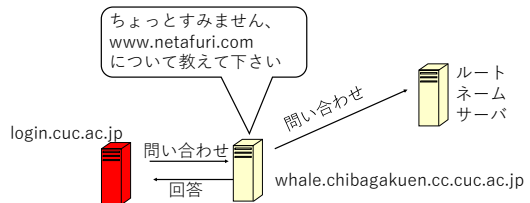
- 実際には13台以上ある
 - 13台のように見えるけど、それぞれ「コピー」がいる



35

ルートネームサーバの仕事

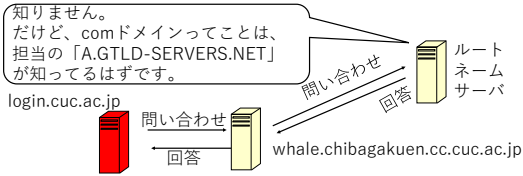
- 世界中からの、名前解決の問い合わせを受ける



36

ルートネームサーバの仕事

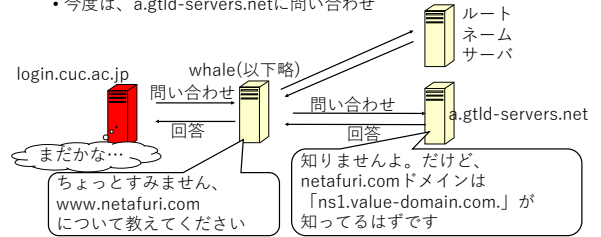
- ルートネームサーバは、問い合わせを受けると、そのトップレベルドメインだけを見て
- 「そのトップレベルドメインの担当」のDNSサーバを提示する



37

この動作をDNSサーバは繰り返す

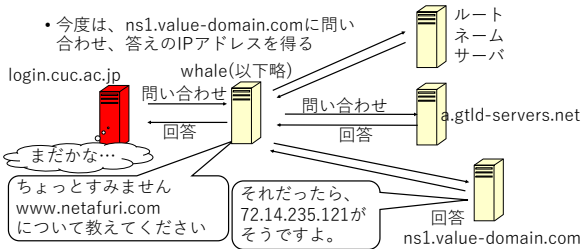
- 今度は、a.gtld-servers.netに問い合わせ



38

更に...

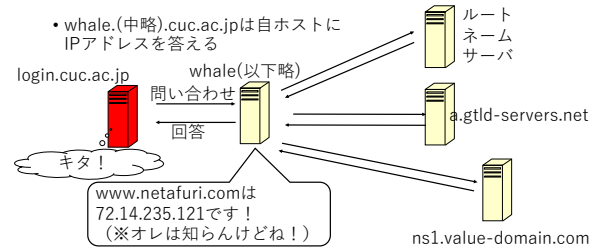
- 今度は、ns1.value-domain.comに問い合わせ、答えのIPアドレスを得る



39

その情報を元に

- whale.(中略).cuc.ac.jpは自ホストにIPアドレスを答える



40

「※オレは知らんけどね」って？

- DNSサーバは「自分が管理管轄している情報」は責任を持って答える
- 自分が管理していない情報は「無責任に」答える
 - Non-authoritative answer
 - ※authoritative=権威ある、信頼できる
 - Non-Authoritative answer=「信頼できない答え」

41

CUCの場合...

- whaleは学外の情報は自信なさげに答える

```

[kohya@hedgehog ~]$ nslookup www.wayo.ac.jp
Server:      202.244.38.73
Address:     202.244.38.73#53

Non-authoritative answer:
Name:   www.wayo.ac.jp
Address: 138.91.12.119

[kohya@hedgehog ~]$
  
```

42

このように

- 名前解決は、DNSという仕組みを用いて階層構造状に順番に尋ねていくという方法を使う
- このため、誰かが新規にドメインを取得しても、担当のDNSサーバだけが作業をすれば良いという仕組みになっている

43

これを

- 実際のコマンドを用いてやってみよう
- UNIXで、**nslookup** とだけ打とう
 - 「対話型プロンプト」という、nslookup専用の会話形式モードが始まり、色々問い合わせ可能になる
 - www.cuc.ac.jp と打ってみよう

```
[kohya@hedgehog ~]$ nslookup
> www.cuc.ac.jp
Server:      202.244.38.73
Address:     202.244.38.73#53

Non-authoritative answer:
www.cuc.ac.jp canonical name = bay.cc.cuc.ac.jp.
Name:   bay.cc.cuc.ac.jp
Address: 202.244.38.102
>
```

44

set querytype=NS

- と打ってみよう
- これで、「DNSサーバは何処か」という情報が問い合わせられるようになる

45

この状態で

- 「cuc.ac.jp」と打つと、「cuc.ac.jp」のネームサーバが出て来る

```
[kohya@hedgehog ~]$ nslookup
> set querytype=NS
> cuc.ac.jp.
Server:      202.244.38.73
Address:     202.244.38.73#53

cuc.ac.jp   nameserver = beach.cc.cuc.ac.jp.
cuc.ac.jp   nameserver = ns.tokyo.wide.ad.jp.
cuc.ac.jp   nameserver = river.cc.cuc.ac.jp.
>
```

46

ここから分かること

- river.cc.cuc.ac.jpとbeach.cc.cuc.ac.jpがCUCのDNSサーバをやってる
- でも、ns.tokyo.wide.ad.jpもCUCのDNSサーバをやってるようだ
- 一方、/etc/resolv.confをみると、CUCのDNSサーバは202.244.38.73 (whale.chibagakuen.cc.cuc.ac.jp) 202.244.38.74 (dolphin.chibagakuen.cc.cuc.ac.jp) がやっているようだ？

47

これは、何故？

- /etc/resolv.confに書かれているDNSサーバのIPアドレスは「誰の為のもの？」
- nslookup(の、set querytype=NS)で出てくるDNSサーバのIPアドレスは「誰の為のもの？」
- Let's 推理！

48

/etc/resolv.confのDNSサーバ

- これは、外部のホストについて名前解決をするために使う、自分たち用サーバ
 - 問い合わせの方向は「中から外」
- これを
キャッシュサーバ(フルサービスリゾルバ)と言う

49

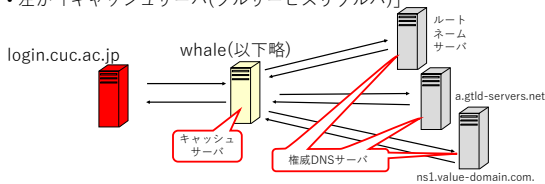
nslookupで出てくるDNSサーバ

- これは、外部の人たちに対して、内部のホスト名の情報を提供するサーバ
 - 問い合わせの方向は「外から中」
- これを
権威DNSサーバ(コンテンツサーバ)と言う
- コンテンツサーバとキャッシュサーバは同じ「ホスト名とIPアドレスを相互に変換する機能」で動いているが、役割は全く違う

50

さっきのこの図で言うと

- 右が「権威DNSサーバ(コンテンツサーバ)」
- 左が「キャッシュサーバ(フルサービスリゾルバ)」



51

おさらい、CUCの話

- www.cuc.ac.jp というサーバが、Webのサービスを行っている、という話はした
- だが、CUC以外の組織が、www.cuc.ac.jp というホストを構築し、サービスを行ってしまう危険性はないのか？

52

ドメインに関する話

- www.cuc.ac.jp を見ていると気づく
- www.keio.ac.jp に似ているのではないかと注：下線部分
- 実は、何処の組織がどういふホスト名を付けることが出来るか、調整している組織がある

53

でも、冷静に考える

- IPアドレスは43億弱、と決まっている
- だが、ドメインは恐らく、無限にあるはずだ！
 - 何故なら、今CUCのWebサーバ名がwww.cuc.ac.jpであるなら…
 - ピリオドで区切られているそれぞれには、アルファベットが任意に入るから！
- ということは…

54

無限にあるドメイン名を

- 管理している組織は超タイヘンだ！
- そこで、管理を楽にする方法が考えられた

55

ドメインの階層構造

- ICANN、という組織がある
- この組織は、ホスト名の一番右側の部分を考える権利を持つ

- `www.cuc.ac.jp`
 - この部分とか
- `www.dameda.net`
 - この部分とか
- `www.netafuri.com`
 - この部分とか

56

具体的に

- ICANNがどういうことをしている組織かはこの講義の範囲外なので割愛
 - 知りたければ調べてください
- だが、ICANNはドメインの「一番右の部分」を、審査の結果増やすことがある
 - 減らすことは、まずない
- この、「ドメインの一番右の部分」を
- 「トップレベルドメイン」という

57

トップレベルドメインの例

- 国(地域)別に持つ、2文字のドメイン
 - .jp(日本)
 - .cn(中国)
 - .uk(イギリス)
 - .tw(台湾)
 - .de(ドイツ)
 - .ch(スイス) その他いっぱい
- こうしたドメインを、ccTLDという
 - ccTLD=カントリーコードトップレベルドメイン

58

トップレベルドメインの例

- 用途の縛りが無い、3文字以上のドメイン
 - .com(元は企業用途)
 - .net(元はネットワークサービス用途)
 - .org(非営利団体)
 - .info(用途不問)
- こうしたドメインを、gTLDという
 - gTLD = ジェネリックトップレベルドメイン
 - gTLDには取得に際する用途の縛りはほとんどない

59

トップレベルドメインの例

- 用途の縛りや制限があり、スポンサーが存在する3文字以上のドメイン
 - .coop(協同組合)
 - .museum(美術館、博物館)
 - .asia(APEC地域の企業用)
 - .tokyo(東京用)
- こうしたドメインを「スポンサードトップレベルドメイン」や「制限付きトップレベルドメイン」などと呼ぶ

60

今のトップレベルドメインの種類

- ジェネリックトップレベルドメイン (gTLD)
- 制限付きジェネリックトップレベルドメイン
- 国別コードトップレベルドメイン (country code top-level domain/略称: ccTLD)
- 国際化ドメイン (internationalized top-level domain/略称: IDN)
- インフラ用トップレベルドメイン (.arpa)

61

権利の例 1

- セカンドレベルドメインを規定する権利
 - 日本の場合、jpより前に、組織の種類に応じてセカンドレベルドメインをつけた
 - ac.jp (学術、研究機関)
 - ed.jp (高等学校)
 - co.jp (企業、営利団体)
 - ne.jp (ネットワークサービス)
 - tokyo.jp (東京の団体)※
 - chiba.jp (千葉の団体 (市役所とか))※

63

権利の例 2

- .comというトップレベルドメインを管理する権利を持つ Verisignという会社は、.comの前 (セカンドレベルドメイン部分) を相手に関わらず自由に、安く売れるようにした
 - dell.comとか
 - ntt.comとか
 - netafuri.comとか
- これで、.comというドメインは爆発的にヒットした
 - 1995~1998年ごろの話

65

それぞれのTLDは

- 管理が相応しいと思われる団体に、管理を委譲する
- .jpの場合は、JPRSという団体
- 従って、JPRSは「???」というドメインであるなら、どんなドメインでも作って良い権利を持つ
 - また、ドメイン作成時のポリシーもコントロールできる

62

余談

- .jpの管理がJPNICからJPRSに移るときに、セカンドレベルドメインに任意の文字を使える「汎用jpドメイン」のサービスが開始された
 - この流れを受けて、tenno.chiyoda.tokyo.jpとか hakkenden.ichikawa.chiba.jpみたいな「地域型JPドメイン (4レベルまであるやつ)」は新規受付を終了した
 - hohogehoge.chiba.jpみたいな、第3レベルに自由な文字列を入れた都道府県型ドメインの取得は可能
- その結果、cuc.ac.jpやcuc.co.jpではなく、cuc.jpみたいなドメインを取得することが可能になった

64

こうした権利をもつ会社を

- レジストリ、と言う
- ただし、レジストリが自分の管轄下の全ドメインの販売管理業務をやっているわけではない
- 販売を、他社に委託することがある
- こうした委託を受けた会社は、レジストラと呼ばれる

66

レジストリとレジストラ

- レジストラは、販売代理店
- レジストリは、販売代理店からの情報を元に登録業務を行う組織
- 実際にドメイン（の権利）を買うのは、レジストラから

67

僕らに出来ること

- 我々一般庶民は、レジストラにお金を払い、任意の名前のドメイン名を利用する権利を一定期間買う
 - 例えば、たかさきは
 - 4.ag 9年間で742ドル
 - damedana.net 1年間で1393円
 - netafuri.com 10年間8千円
 - などを保有している
 - レジストラやトップレベルドメインによって値段は違う
 - 他にも複数所有

68

ただし

- .comドメインに関しては、もう1単語のドメインは取得不可能だと思っている
 - takasaki.comとかはもう他の人が持っている
 - cuc.comも然り
 - chiba.comとかも駄目
 - 世界中の人が先を争って取得したからだ！
- でも、ドメインを取得だけして使わずにいる組織も山ほどある

69

休眠ドメイン

- ブラウザで `http://www.????.com/` と入力して、Webページが出てこないドメインでも、既に誰かに取得されているかもしれない
 - Registeredされているけどconnectedではない
 - ていうかそれをやっちゃ駄目！
 - それをやってはいけない理由はなに？推理！
- そういうドメインを、どうやって調べる？

70

レジストラ

- お名前.comやMuuMuuドメイン、Valueドメインなど日本にはドメインを販売する会社が複数ある
- 日本国外のドメインすら取得可能
- そういう会社のサイトでは、そのドメインが既に取得されているか否かを調べることができる
- あるいはwhois検索というサービスを公開してくれているサイトで調べてみる

71

一例

- <https://tech-unlimited.com/> のwhois検索で takasaki.comを検索した結果
- 当然(誰か知らない人に)取得されている
- 
 - No match と言われれば未取得である

72

ドメインの多言語化

- ドメイン名にアスキー文字以外を使いたい！というニーズがもともとあり、punycodeという「Unicodeのエンコード方法」を用いてドメイン部分に英語以外の言語を表記可能になった
- トップレベルドメインにおいては、日本語では「みんな」というドメインが実用化している
 - セカンドレベルドメイン以下で多言語に対応するかどうかは、そのドメインを預かるレジストリが勝手に決めてよい
 - .comは多言語化されており、僕も「井.com」というドメインを持っている

73

ドメインと信用

- 仮にドメインを取得する場合
ドメインによって信用の格が違ってくる
- 誰にでも簡単に取得できるドメインはドメインとしての信用されにくい
 - 安いドメインも同様
 - ccTLDでも、その国に住所を持っていなくても自由に取得可能なドメインは信用されにくい
- 取得してすぐのドメインも信用されにくい
 - whoisで「いつ取得したか」は簡単にわかる

74

日本においては

- .co.jp ドメインの信頼性は群を抜く
- その分取得にかかる手間暇も大きい

75

というわけで本日のお題

- <https://forms.gle/xaiVutUQVv1ihbu46>

76