## EXAMEN: Computergebruik

1e Bachelor Informatica prof. dr. Peter Dawyndt groep 3 vrijdag 05-09-2014, 8:30 academiejaar 2013-2014 tweede zittijd

Opgave 1 (15 pt)

Gebruik filters, I/O redirection en pipes om telkens een commando samen te stellen dat uitvoer genereert conform onderstaande beschrijvingen. Hierbij is het toegelaten om gebruik te maken van sed, maar niet van andere programmeerbare filters zoals awk, perl, .... Vermijd dat de commando's (tijdelijke) bestanden aanmaken binnen het bestandssysteem, tenzij dat expliciet gevraagd wordt.

1. De website randomfunfacts.com toont je bij elk bezoek een interessant weetje, dat je volgens de website kunt gebruiken om indruk te maken op je vrienden.



Het Unix commando cowsay (beschikbaar op helios) kan onder andere gebruikt worden om een koe iets te laten zeggen.

Gebruik bovenstaande informatie om een commando samen te stellen dat Tux een weetje uit de randomfunfacts.com databank laat zeggen. Voor het weetje uit bovenstaande figuur moet dan bijvoorbeeld het volgende gegenereerd worden op standaard uitvoer:

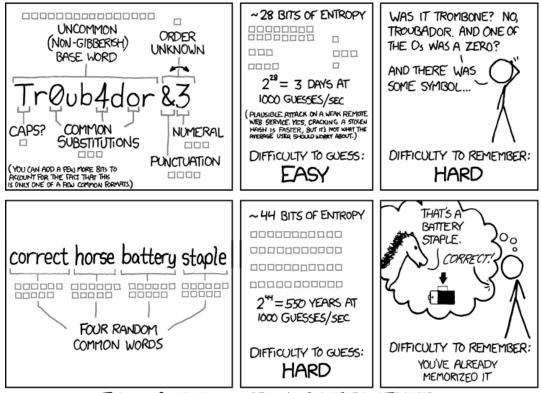
2. In de Unix-wereld worden verschillende soorten bestanden gebruikt. Elk type wordt voorgesteld door één enkele letter: f voor gewone bestanden, d voor directories, s voor symbolische links, p voor benoemde pijpleidingen (named pipes), s voor sockets, b voor blok-gebaseerde device files en c voor karakter-gebaseerde device files.

Schrijf een bash shell script bestandstypes waaraan de padnaam van een directory moet doorgegeven worden. Het script moet een overzicht weergeven van het aantal bestanden per type die onder deze directory (dus inclusief alle subdirectories) worden teruggevonden. In dit overzicht moeten de types opgelijst worden in dalende volgorde van het aantal gevonden bestanden van

dat type. Voor de bestanden onder de /dev directory moet bijvoorbeeld het volgende overzicht gegenereerd worden:

```
$ bestandstypes /dev
237 1
174 c
35 b
20 d
1 f
1 s
```

3. Onderstaande cartoon stelt een manier voor om willekeurige wachtwoorden te genereren die je makkelijk kunt onthouden



THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Vul de commandolijn uit onderstaande sessie aan, zodat deze een willekeurig wachtwoord uitschrijft naar standaard uitvoer dat opgebouwd is op basis van de volgende procedure:

- (a) vertrek van een willekeurige woordenlijst die opgeslagen zit in een tekstbestand dat één enkel woord per regel bevat (bijvoorbeeld /usr/share/dict/words op helios)
- (b) kies uit het tekstbestand willekeurig vier woorden die bestaan uit vier tot acht letters
- (c) plak de gekozen woorden in camelCase achter elkaar: hierbij moet de eerste letter van het eerste woord dus omgezet worden naar een kleine letter, en de eerste letter van alle volgende woorden moet omgezet worden naar een hoofdletter

Onderstaande sessie bevat alvast een aantal willekeurige paswoorden die we op deze manier gegenereerd hebben:

```
$ cat /usr/share/dict/words | ...
browsersHemdjesLangbeenPokerden
$ cat /usr/share/dict/words | ...
databitsStoverijBlankenMeepakt
$ cat /usr/share/dict/words | ...
mengtaalErkertjeMankpootAlgoede
```

4. De idee om een zin — of een langer stuk poëzie of proza — te schrijven waarvan de lengtes van de opeenvolgende woorden de cijfers van het getal  $\pi$  (=3.14159265358979...) vormen, is ontstaan ergens begin 1900. Eén van de vroegste en bekendste voorbeelden is de volgende zin, die vermoedelijk werd opgesteld door de Engelse fysicus Sir James Jeans:

How I need a drink, alcoholic in nature, after the heavy lectures involving quantum mechanics! Het eerste woord van de zin bestaat uit drie letters, het volgende woord uit één letter, het volgende woord uit vier letters, enzoverder, en samen vormen ze de eerste vijftien cijfers van het getal  $\pi$ . Een iets langer voorbeeld is het gedicht met ABAB rijmschema uit het boek Playing With Words uit 1960 van de hand van Joseph Shipley.

But a time I spent wandering in bloomy night; Yon tower, tinkling chimewise, loftily opportune. Out, up, and together came sudden to Sunday rite, The one solemnly off to correct plenilune.

In dit zogenaamde Pilish stelt elk woord van n letters

- (a) het cijfer n voor als n < 10
- (b) het cijfer 0 voor als n = 10
- (c) de opeenvolgende cijfers van n voor als n > 10 (een twaalfletterwoord stelt bijvoorbeeld de cijfers 1 en 2 voor)

Hierbij is regel (c) eigenlijk geen speciaal geval, maar is het regel (b) die de uitzondering vormt als we het algoritme op de volgende manier omschrijven: om de cijfers van het getal  $\pi$  te bepalen aan de hand van een tekst die geschreven is in Pilish, schrijf je het aantal letters van het woord naast elk woord (behalve bij tienletterwoorden waar een nul wordt naast geschreven). Daarna lees je alle cijfers in volgorde van begin tot eind om zo de waarde van  $\pi$  te bekomen.

Om helemaal eenduidig te zijn, moeten we nog vastleggen hoe leestekens geïnterpreteerd worden in een tekst in Pilish — of meer algemeen, hoe elk symbool moet behandeld worden dat geen letter is (A-Z of a-z). De regels die we zullen gebruiken zijn de volgende:

- (a) Als een woord één of meer aanhalingstekens bevat, dan worden deze gewoon genegeerd. Het tekstfragment couldn't wordt daardoor behandeld alsof er couldnt stond, wat een woord van zeven letters oplevert.
- (b) Elk karakter dat geen letter of aanhalingsteken is, wordt behandeld als een scheidingteken. Dat komt op hetzelfde neer als zeggen dat dergelijke karakters op dezelfde manier behandeld worden als witruimte.

Een gevolg van de tweede regel is dat woorden met een koppelteken — zoals fun-filled — worden behandeld als twee afzonderlijke woorden, waardoor ze minstens twee cijfers genereren (in dit geval 3 en 6). Op basis van deze regels genereren leestekens nooit cijfers als de tekst naar cijfers wordt omgezet. Veronderstel dat we bijvoorbeeld fun and games willen schrijven, maar dat de volgende twee cijfers van  $\pi$  3 en 5 zijn. Dan kunnen we het gewenste resultaat bekomen door fun & games te schrijven, omdat de ampersand genegeerd wordt bij het omzetten van tekst naar cijfers.

Schrijf een bash shell script pilish waaraan een zin als argument moet meegegeven worden. Het script moet de reeks cijfers naar standaard uitvoer uitschrijven die bekomen wordt als men de zin vertaalt volgens de regels van het Pilish. Onderstaande sessie bevat enkele voorbeelden van een dergelijke vertaling.

```
$ pilish "But a time I spent wandering in bloomy night;"
314159265
$ pilish "Yon tower, tinkling chimewise, loftily opportune."
358979
$ pilish "Out, up, and together came sudden to Sunday rite,"
323846264
$ pilish "The one solemnly off to correct plenilune."
3383279
```

Opgave 2 (10 pt)

Het tekstbestand WK2014.txt bevat informatie over alle spelers die voorkwamen op de officiële ploegenlijst van de FIFA voor de wereldbeker voetbal 2014 in Brazilië. Dit bestand bevat een hoofding van één regel, gevolgd door telkens één regel informatie per speler. De informatievelden worden gescheiden door een komma waar eventueel spaties rond staan. Elke informatieregel bevat de volgende velden: i) naam, ii) land, iii) rugnummer, iv) positie, v) geboortedatum, vi) aantal gespeelde interlands en vii) ploeg.

Gevraagd wordt om — gebruik makend van de teksteditors vi of vim — een reeks commando's op te stellen die achtereenvolgens de volgende opdrachten uitvoeren. Probeer voor elke opdracht zo weinig mogelijk commando's te gebruiken en zorg er voor dat elk van deze commando's bestaat uit zo weinig mogelijk tekens. Alle opdrachten moeten na elkaar uitgevoerd worden. Ter controle kan je gebruik maken van de meegeleverde bestanden WK2014.i  $(1 \le i \le 5)$ , die telkens de inhoud van het bestand bevatten nadat de i-de opdracht werd uitgevoerd.

1. Verwijder de hoofding en eventuele spaties aan het begin en einde van elke regel. Bijvoorbeeld:

```
#naam, land, rugnummer, positie, geboortedatum, aantal gespeelde interlands, ploeg
Alireza Jahanbakhsh, Iran, 9, FW, 1993-08-11, 7, NEC

Marcelo Brozovic, Croatia, 14, MF, 1992-10-16, 1, Dinamo Zagreb
Claudio Bravo, Chile, 1, GK, 1983-04-13, 79, Real Sociedad

Wordt omgezet naar

Alireza Jahanbakhsh, Iran, 9, FW, 1993-08-11, 7, NEC
Marcelo Brozovic, Croatia, 14, MF, 1992-10-16, 1, Dinamo Zagreb
Claudio Bravo, Chile, 1, GK, 1983-04-13, 79, Real Sociedad

Claudio Bravo, Chile, 1, GK, 1983-04-13, 79, Real Sociedad
```

2. Vervang de komma als veldscheidingsteken door een verticale streep (|) en zorg ervoor dat er exact één spatie voor en na elk veldscheidingsteken staat. Toegepast op het vorige voorbeeld wordt dit

```
| Alireza Jahanbakhsh | Iran | 9 | FW | 1993-08-11 | 7 | NEC
| Marcelo Brozovic | Croatia | 14 | MF | 1992-10-16 | 1 | Dinamo Zagreb
| Claudio Bravo | Chile | 1 | GK | 1983-04-13 | 79 | Real Sociedad
| . . .
```

3. Verander de velden van plaats zodat dit de nieuwe volgorde wordt: i) geboortedatum, ii) rugnummer, iii) land, iv) ploeg, v) positie, vi) naam en vii) aantal gespeelde interlands. Toegepast op het vorige voorbeeld wordt dit

```
1 | 1993-08-11 | 9 | Iran | NEC | FW | Alireza Jahanbakhsh | 7
2 | 1992-10-16 | 14 | Croatia | Dinamo Zagreb | MF | Marcelo Brozovic | 1
3 | 1983-04-13 | 1 | Chile | Real Sociedad | GK | Claudio Bravo | 79
4 | ...
```

4. Duid alle doelmannen (positie = GK) aan door die regel met een ampersand (&) te laten beginnen. Toegepast op het vorige voorbeeld wordt dit

```
1 | 1993-08-11 | 9 | Iran | NEC | FW | Alireza Jahanbakhsh | 7 | 2 | 1992-10-16 | 14 | Croatia | Dinamo Zagreb | MF | Marcelo Brozovic | 1 | 3 | & 1 | Chile | Real Sociedad | GK | Claudio Bravo | 79 | 4 | ...
```

5. Sorteer de regels omgekeerd alfabetisch volgens het land van herkomst. Bij een gelijk land moet er oplopend gesorteerd worden op rugnummer. Toegepast op het vorige voorbeeld wordt dit

```
1 | &1986-06-16 | 1 | Uruguay | Galatasaray | GK | Fernando Muslera | 58

2 | 1980-11-02 | 2 | Uruguay | West Bromwich Albion | DF | Diego Lugano | 94

3 | 1986-02-16 | 3 | Uruguay | Atletico Madrid | DF | Diego Godin | 77

4 | ...
```

Opgave 3 (10 pt)

Een QR-code is een tweedimensionale streepjescode die in 1994 werd ontwikkeld door Denso Wave, een dochteronderneming van het Japanse bedrijf Denso. De letters QR zijn een afkorting van Quick Response (snel antwoord). QR-codes zijn vierkant en worden opgebouwd uit zwarte en witte vierkante blokjes. Daardoor kunnen QR-codes ook makkelijk tekstueel voorgesteld worden. In deze opgave worden witte blokjes van een QR-code voorgesteld door spaties, en zwarte blokjes door hekjes (#).



Het tekstbestand qr.txt bevat een aantal QR-codes. Elke regel van het bestand correspondeert met één enkele rij van een QR-code, en wordt gevolgd door een verticale streep (|) en een woord dat enkel bestaat uit kleine letters. Gevraagd wordt:

- 1. Bepaal reguliere expressies voor elk van onderstaande verzamelingen. Daarbij staat  $\mathcal{Q}$  voor de verzameling van alle mogelijke rijen van QR-codes (strings die enkel bestaan uit spaties en hekjes). Probeer de reguliere expressies bovendien zo kort mogelijk te houden.
  - (a)  $\alpha = \{q \in \mathcal{Q} \mid \text{reeksen hekjes worden telkens van elkaar gescheiden door één enkele spatie}^1\}$ 
    - <sup>1</sup> voor de eerste reeks hekjes en na de laatste reeks hekjes mogen dus een willekeurig aantal spaties voorkomen

- (b)  $\beta = \{q \in \mathcal{Q} \mid q \text{ bevat een patroon van lengte 6 dat minstens drie keer voorkomt}^2\}$ 
  - <sup>2</sup> herhaalde voorkomens van het patroon mogen elkaar niet overlappen

```
voorbeelden: ###__##__##__##__##__###\beta,
```

(c)  $\gamma = \{q \in \mathcal{Q} \mid \text{reeksen hekjes hebben altijd een even lengte} \}$ 

```
voorbeelden: ##..###.##..##..##..##..##...# \neq \gamma
```

(d)  $\delta = \{q \in \mathcal{Q} \mid \text{ lengte van opeenvolgende reeksen hekjes is dalend}^3 \}$ 

Gebruik een commando uit de grep familie om enkel die regels van het bestand qr.txt te selecteren, waarvan de QR-coderij behoort tot de opgegeven verzameling. Vermeld in je antwoordbestand voor elke verzameling het gebruikte selectiecommando, en geef telkens ook aan hoeveel regels je gevonden hebt.

- 2. Beschouw de verzamelingen  $\alpha$ ,  $\beta$ ,  $\gamma$  en  $\delta$  zoals hierboven gedefinieerd. Gebruik nu deze verzamelingen om op de volgende manier een boodschap bestaande uit vier woorden te achterhalen:
  - (a) het eerste woord staat op de unieke regel met een QR-coderij uit de verzameling  $\alpha \cap \beta$
  - (b) het tweede woord staat op de unieke regel met een QR-coderij uit de verzameling  $\beta \cap \gamma$
  - (c) het derde woord staat op de unieke regel met een QR-coderij uit de verzameling  $\gamma \cap \delta$
  - (d) het vierde woord staat op de unieke regel met een QR-coderij uit de verzameling  $\delta \cap \alpha$

Vermeld in je antwoordbestand de gevonden woorden, samen met het Unix commando (of de commandosequentie) dat je gebruikt hebt om elk van deze woorden te vinden.

Geef LATEX-code die precies hetzelfde resultaat oplevert als het tekstfragment in onderstaand kader. Zorg ervoor dat de opmaak zo getrouw mogelijk behouden blijft. Definieer hiervoor een LATEX-macro om de tabelcellen met inhoud van de vorm  $\left(\frac{a}{b}\right)^{\frac{c}{a}}$  makkelijk te kunnen invullen. Plaats een naar PDF gecompileerde versie van het LATEX-fragment in het ZIP-bestand dat je indient via Indianio.

Het boek *Mathematics in the Making* van Lancelot Hogben bevat een tabel waarmee je de basis goniometrische waarden makkelijk kunt onthouden.

	0°	30°	45°	60°	90°
sin	$\left(\frac{0}{4}\right)^{\frac{1}{2}}$	$\left(\frac{1}{4}\right)^{\frac{1}{2}}$	$\left(\frac{2}{4}\right)^{\frac{1}{2}}$	$\left(\frac{3}{4}\right)^{\frac{1}{2}}$	$\left(\frac{4}{4}\right)^{\frac{1}{2}}$
cos	$\left(\frac{4}{4}\right)^{\frac{1}{2}}$	$\left(\frac{3}{4}\right)^{\frac{1}{2}}$	$\left(\frac{2}{4}\right)^{\frac{1}{2}}$	$\left(\frac{1}{4}\right)^{\frac{1}{2}}$	$\left(\frac{0}{4}\right)^{\frac{1}{2}}$
tan	0	$3^{-\frac{1}{2}}$	1	$3^{\frac{1}{2}}$	$\infty$

De onderste rij kan afgeleid worden uit het feit dat  $\tan \theta = \frac{\sin \theta}{\cos \theta} = \cot \left(\frac{\pi}{2} - \theta\right) = \frac{1}{\cot \theta}$ .

 $<sup>^3</sup>$ niet noodzakelijk strikt dalend, m.a.w. ope<br/>envolgende reeksen hek<br/>jes met gelijke lengte zijn toegelaten