Tilpro Övning 1 2006

Per-Anders Staav (vikarierande)
(föredrar att kallas Pa)
Kan nås via epost: pasta@kth.se

På programmet idag:
Unix (Solaris)
Grunderna i Python programmering

Allmänt om datorkonton

*Nadas konton är inte samma som Kth konton. Besök Delphi och fixa konto innan labben
*Era passerkort fungerar till datorsalarna
*Testa att ert konto fungerar innan labben, vi kommer i labbsalen endast kunna hänvisa till Delphi
*Köp unix häftet från Delphi, den innehåller massor nyttigt som ni behöver när ni ska labba
*När ni ska logga in. Klicka på Option och sedan session och se till att ni väljer valet CDE. Övriga fönsterhanterare fungerar ibland sämre.

Kommandot ls

Kommando: ls
Effekt: listar filer i foldern

Kommando: ls -a
Effekt: listar även gömda filer i foldern

Kommando: ls -l
Effekt: listar utförlig information om filerna i foldern

Kommandot cd

Kommando: cd bibliotek
Effekt: flyttar dig in i biblioteket

Kommando: cd ..
Effekt: flyttar dig tillbaka till biblioteket ovanför

Kommando: cd ~
Effekt: flyttar dig tillbaka till ditt hembibliotek

Sökvägen till hembiblioteket är något i stil med
/afs/nada.kth.se/home/n/u1004mnm/

afs

Ett system som heter afs gör att du kan läsa dina filer oavsett dator du sätter dig vid.

Vanligtvis pekar ~<personsanvändarnamn>/ på hemkatlogen för personen. Vill man hitta sökvägen till ett konto men inte vet användarnamn för kan man pröva kommandot finger <persons namn>

Programmet KerberosFtp låter dig logga in på nada hemifrån och lägga upp och ta ned filer.

Kommando cp

Kommando: cp filAttKopiera filAttKopieraTill
Effekt: kopierar filen

Ofta byter man målfilen till bokstaven . som innebär att filen läggs i biblioteket där man står

Man kan använda * för att kopiera en mängd filer.
*.py betyder alla filer som slutar med ändelsen py

man

Kommando: man sort
Effekt: visar en text sida med beskrivningar om hur kommandot sort fungerar...alla kommandon har dock inte hjälpsidor som går att hitta automatiskt

Finns massor av nyttiga unix kommandon jag inte hinner gå igenom. Kolla gärna upp vad de gör t.ex. sort, grep, cat

Utskrifter

Kommando: lpq
Effekt: Visar vilka utskrifter som skrivs på skrivaren

Kommando: lprm -
Effekt: tar bort dina egna utskrifter

Kommando: lprm <nummer>
Effekt: tar bort utskrift med detta nummer

Kommando: setprinter brun
Effekt: Gör så att skrivaren som heter brun blir default skrivare för detta terminal fönster.

Utskrift forts

Kommando: lpp
Effekt: skriver ut textfiler OBS skriver du ut pdf-fil eller bild med denna får du ut 100 tals sidor med skräptecken. Stoppa sådan utskrift omedelbart om den råkat starta.

Kommando: a2ps
Effekt: skriver ut programfiler lite snyggare

Kommando: lp
Effekt: Skriver ut filer...läs noggrant i man sidan innan ni prövar att skriva ut

Dela konto

Kommando: fs lq
Effekt: Visar hur mycket plats du har kvar på kontot

Kommando: fs la tilpro
Effekt: Visar information om vilka som får titta och göra saker med tilpro

Kommando: fs sa tilpro <kompis> rlidwk
Effekt: Låter person användarnamn kompis komma åt tilpro. Skapar man filer i underkataloger till tilpro kan kompisen komma åt dem med

Dela konto forts

Kommando: ln -s /afs/nada.kth.se/home/n/u1004mnm/tilpro tilpro
Effekt: Gör att tilpro blir en länk som vi kan gå in med cd. Gör vi dock cd .. från den katalogen hamnar vi kompisens hembibliotek. Raderas länken påverkas inte filerna som ligger i biblioteket.

-s ovan betyder att det är en symbolisk länk. Detta innebär att målet kan ligga på annan hårddisk. På nada behöver man alltid göra det eftersom man inte vet vilken server filen hamnar på.

rm

Kommando: rm slask.py
Effekt: tar bort filen slask.py Det går att använda * för att ange flera filer men det rekomenderas ej. Raderar man för många filer är det osäkert om de alls går att få tillbaka (de kan dock ligga backupade i foldern OldFiles)

Mkdir och rmdir

Kommando: mkdir tilpro
Effekt: Skapar ett ny bibliotek i foldern du befinner dig.

Kommando: rmdir slaskFolder
Effekt: Tar bort foldern förutsatt att den är tom

allmänt

Alla kommandon körs enklast från terminal fönstret.
Man måste ändå köra python programmen därifrån

Kommando: emacs &
Effekt: Startar editorn emacs i bakgrunden så att terminalfönstret inte hänger sig

Kommando: sm
Effekt: Startar hjälpsystemet så att du kan köa för hjälp. Se till att ni väljer rätt kurs...

Kommando: python fil.py
Effekt: kör programmet som finns i filen fil.py

Moduler

Kommando: module add python/latest
Effekt: Gör att du i terminalfönstret kan använda senaste python versionen

Vill man se till att viss modul alltid laddas kan man lägga till kommandot till filen .modules in din hemkatalog. Punkten framför filnamnet betyder att den bara synns med ls om man lägger till -a växeln. Filen går dock utmärkt att editera i emacs

Annat nyttigt

För att skapa ny fil i emacs. Välj open file och skriv fil som inte finns. Filen skrivs dock inte till hårddisken förren du trycker på spara.

Firefox är bra webläsare från modulen firefox som du kan starta med firefox &

Sima programmet som startas med kommandot sm finns i modulen sima

Det finns fyra skrivbord man kan lägga sina fönster på. Klicka på ruta i nedre delen av skärmen för att byta skrivbord.

Python programmering

Att skriva program är lätt...att skriva program som gör rätt sak är svårare....att veta säkert att programmet gör rätt sak är enormt svårt.

Python program går snabbt att skriva men kan vara svårt att hitta fel i. Testar man saker noggrant hittar man dock alla fel och med python hinner man testa mycket.

Vilka byggblock finns i Program?

Inmatning – Att vi hämtar data från tangentbord eller filer

Utmatning – Skriver data till skärmen (eller något annat lämpligt)

Variabler – Lagrar data som programmet arbetar med

Vilkorssatser – Gör att vi kan välja vilken kod som ska köras beroende på vad som finns i en variabel

Slingor – Låtor oss göra en uppsättning rader i programmet flera gånger

...dessutom metoder från funktionsbibliotek

Exempel på Inmatning till variabel

variabel=**input**(“Ange vad du vill lägga in i variabeln”)

Input används för att ta emot inmatning och direkt översätta till siffra. Det som kallas variabel ovan får automagiskt lämplig sort för att kunna ta emot just en siffra. Hade man använt annan sorts inmatning hade variabel fått annan sort (mer om inmatning senare).

Villkorssatser + utmatning

```
variabel=input(“Ange vad du vill lägga in i
  variabeln”)
if variabel > 0 :
    print“Du angav ett positivt tal”
else:
    print“Talet är troligen negativt...”
```

Om villkoret är sant körs den första print raden. Är det falskt körs det som står efter else (mer om villkorssatser senare)

Slingor

```
variabel=input()
while variabel < 30:
    print“Ett nytt varv i slingan”
    print variabel
    variabel = variabel +1
```

En slinga är en sorts villkorssats som körs om och om igen så länge villkoret är sant. Det som upprepas är de rader som följer som har samma inflyttning som raden som följer while raden. I python får man bara flytta in raden om föregående rad slutade med tecknet : (mer om slingor senare)

Utskrift till skärmen

print“Proxxi”
print“är bra”

Proxxi
är bra

print“Proxxi”,
print“är bra”

Proxxi är bra

Kommentarer

#Kommentarer används för att förklara obegripliga
#kodrader. När programmet körs så hoppar python
#över det som följer tecknet #

#Det finns dock special fall....följande är bra att
#skriva om man vill ha svenska tecken
#coding: iso-8859-1

print“Hejsan Världen”

Variabelnamn

Variabelnamn måste börja med en amerikansk bokstav, i resten av namnet får det finnas amerikanska bokstäver, siffror samt tecknet _

Viktigt att komma ihåg är att det skillnad på stora och små bokstäver. Variabelnamnen proxxi_data och Proxxi_data pekar alltså på olika variabler!

Alltså använd ej å,ä eller ö i variabel namn!

Variabelsorter

För att man ska kunna räkna på saker är vissa variabler inte bara tecken
Siffor:
    *integer
    *long
    *float
    *complex number (imaginära enheten heter j)
*String (ej ändringsbar)
*List
*Tuple (ej ändringsbar)
*Dictionary (mappar söknycklar till dataposter)

Reserverad ord i Python

Vissa ord ingår i själv Python syntaxen och får därför inte vara variabelnamn:

and, assert, break, class, continue, def, del, elif, else, exceptif, exec, finally, for, from, global, or, import, if, in, is, lambda, not, return, pass, print, raise, try, while

Omvandling mellan typer

String till int:
agetext = “20”
age = **int**(agetext)

Int till string:
weekdays = 7
wdtext = **str**(weekdays)

For slingor

For tempvariabel in namn_lista:
    Satser man vill ska upprepas flera gånger

Exempel

```
for iter in [-1,'a',191,”glurg”]:
    print“Hurra”, iter,
```

Hurra -1 Hurra a Hurra 191 Hurra glurg

```
for iter in range(4):
    print“Hurra”, iter,
```

Hurra 0 Hurra 1 Hurra 2 Hurra 3

Operatorer

+,-,* och / fungerar som vanligt för tal
Om det är heltal ger 12 % 9 siffran 3 som svar (dvs det är modulo operatorn från matematiken)
2**5 returnerar 2*2*2*2*2 (dvs 32)

* och + kan också användas på listor och strängar

```
tal_lista=[1,2,3]
2*tal_lista
```

1 2 3 1 2 3

```
tal_lista=[1,2,3]
tal_lista + [5,6]
```

1 2 3 5 6

Jämförelse operatorer

Vill man jämföra två saker använder man någon av följande operatorer

== (de innehåller samma sak)

is (de pekar på samma sak)

!= (de innehåller ej samma sak)

not (de pekar ej på samma sak)

<

<=

>

>=

(is och not används normalt ej)

Funktioner

Vill man vid flera tillfällen i programmet göra samma sak så kan lägga den koden i en funktion. En funktion ser ut på följande vis.

```
def exempel_funktion(indata,annan_indata):
    #här följer kod som gör de viktiga som funkionen
    #ska lösa. T.ex. säger om indata är ett primtal
```

Önskad effekt av vårt första program

exempelkörning:
Ange vad nuvarande euro kurs är: 8.95
100 kr är 11.173184 stycken euro

8.95 är inmatat av användaren...talet 11.17 osv beräknas av programmet och stoppas in på rätt ställe

Ett första försök till lösning

```
#coding: iso-8859-1
print ”Ange vad nuvrande euro kurs är:”,
euro_kurs=input()
euro=100/euro_kurs
print “100 kr är”
print euro
print “stycken euro”
```

exempelkörning:

Ange vad nuvarande euro kurs är: 8.95
100 kr är
11.173184
stycken euro

Fixar radbrytning genom att lägga till komma efter vissa print rader

```
#coding: iso-8859-1
print ”Ange vad nuvrande euro kurs är:”,
euro_kurs=input()
euro=100/euro_kurs
print “100 kr är”,
print euro,
print “stycken euro”
```

exempelkörning:
Ange vad nuvarande euro kurs är: 8.95
100 kr är 11.173184 stycken euro

Om användaren ger euro kursen som heltal kommer inga decimaler

```
#coding: iso-8859-1
print ”Ange vad nuvrande euro kurs är:”,
euro_kurs=input()
euro=100.0 / euro_kurs
print “100 kr är”,
print euro,
print “stycken euro”
```

exempelkörning:
Ange vad nuvarande euro kurs är: 8
100 kr är 12.5 stycken euro

Exempelprogram 2

```
#coding: iso-8859-1
svar=raw_input(”Vad är lösen?”)
while( svar != “All makt åt Tengil vår befriare”):
    print “Odugling...ange lösen”
    svar=raw_input()
print“Ja...då kan du passera”

#Ett logiskt skjysst program....fast det är inte
#säkert att terminal vi kör på låter oss skriva
#å,ä eller ö så att python förstår det
```

While vilkoret ligger alltid först...

While inleds alltid med ett test...i verkligheten vill man ofta ha sekvensen fråga, inmatning och sist test.

En lösning är att lägga till en extra fråga och inmatning innan while slingan startas.

En annan möjlig lösning är att man lägger
svar=”dummy”
Sekvensen blir då egentligen test, fråga inmatning test, fråga....men det lider ju inte användaren av

Exempel program 3

```
def bedom(kulgrad):
    if kulgrad < 0:
        print”Man kan väl inte ha negativt kul!!!”
    elif (kulgrad == 0):
        print”Det ändrar sig nog när du börjat”
    elif kulgrad > 100:
        print”Precis rätt attetyd”
    elif kulgrad == 159:
        print ”En utskrift som aldrig skrivs ut”
    else:
        print”Bra, kan kanske bli bättre?”
print”Hur pass kul tror du python är?”
bedom(int(raw_input(“(Ange ett siffervärde):”)))
```

Exempel program 4

```
#coding: iso-8859-1
#Mata in ett tal: 4
#1 2 3 4
#2 4 6 8
#3 6 9 12
#4 8 12 16
tal =input("Mata in ett tal:")
for i in range(1,(tal+1)):
    for j in range(1,(tal+1)):
        print (i*j),
    print
```

Uppmaning

Skaffa python kramaren och skriv massor av test program. Man lär sig genom att skriva program.

Öva att lägga till fel i fungerande kod för att se vilken sorts fel ni får.

Nyttja länkarna ni kan nå via kurshemsidan.

Lycka till på labben