

linux: Prosessit

linux: Prosessit

- ▶ Jokainen komento käynnistää vähintään yhden **prosessin**
- ▶ Jokaiselle prosessilla **tunniste PID**, jolla prosessiin voidaan viitata.
- ▶ Jokaisella prosessilla on **prioriteetti PRI**. Se kertoo paljonko prosessi käyttää koneen resursseja muihin prosesseihin nähden
- ▶ Normaalisti komento käynnistää prosessin **aktiiviseksi**
- ▶ **komento &** käynnistää prosessin **taustalle**
- ▶ **Esimerkki: emacs nimi &** siirtää editorin **taustalle** ja **linux** komentotulkissa **voi** edelleen antaa muita komentoja
- ▶ **ps** tulostaa aktiiviset prosessit
- ▶ **Ctrl** + **c** lopettaa aktiivisen prosessin
- ▶ **kill PID** lopettaa prosessin **PID**, jos siihen on oikeudet

linux: Prosessit

- ▶ **jobs** tulostaa käynnissä olevat työt. Tulostus voi olla vaikkapa
[1] Running emacs tila2015.tex &
[5]- Running evince tila2015.pdf &
- ▶ **top** listaa tietoa käynnissä olevista prosessit prosessorin käytön mukaan
- ▶ Prioriteetti välillä $-20 \geq \text{PRI} \leq +19$.
- ▶ Pienempi **PRI** $\Leftrightarrow$ Tärkeämpi prosessi
- ▶ Käyttäjien **PID** ≥ 0 (Oletus = 0)
- ▶ **nice -n arvo komento** käynnistää komennon prioriteetilla **PRI=arvo**.
- ▶ **Esimerkki: nice -n +3 python** käynnistää **python**:n prioriteetilla **PRI=3**
ps -lx | grep python kertoo, että prosessilla on **PID=6632**
renice +5 6632 muuttaa **PRI=5**:een
- ▶ **PRI** arvoa voi vain kasvattaa (vähentää prosessin tärkeyttä) **Miksi ja milloin?**

String manipulointi

python

- ▶ Useita **python** komentoja

Esimerkki 1

```
A='abcd'; x=A.find('c'); print(x)
```

tulostaa 2

Esimerkki 2

```
B='o,1,p'; x=B.split(','); print(x)
```

tulostaa ['o', '1', 'p']

',' = "Erotin"

Esimerkki 3

```
B='o 1 p'; x=B.split(); print(x)
```

tulostaa ['o', '1', 'p']

tyhjä="Oletus (default) erotin"

- ▶ Mahdotonta muistaa ulkoa vuoden (päivän, tunnin?) päästä
- ▶ **grep .split *.py** ⇒ Malli!

octave

- ▶ Vastaavia **octave** komentoja

Esimerkki 1

```
A='abcd'; x=strfind(A,'c'); disp(x)
```

tulostaa 3

Esimerkki 2

```
B='o,1,p'; x=strsplit(B,','); disp(x)
```

tulostaa

```
{
  [1,1] = o
  [1,2] = 1
  [1,3] = p
}
```

Esimerkki 3 (Kokeile itse)

tyhjä="Oletus (default) erotin"

- ▶ **Esimerkkiohjelmat** seuraavat sivut
Omalli7.m ja **Pmalli7.py**

String manipulointi

```
# Kommenttirivi: Tama on python ohjelmani Pmalli7.py
# – Joitakin esimerkkeja stringien manipuloinnista
#
import os ; os.system('clear')           # Tyhjennetaan naytto
a="karhunkierros" ; b=a.find("karhu")    # Etsi teksti
print('b_=',b)                          # Tulos
d='AbCd' ; e=d.upper()                   # Isoiksi kirjaimiksi
print('d_=',d,'e=',e)                   # Tulos
f='EfGh' ; g=f.lower()                   # Pieniksi kirjaimiksi
print('f_=',f,'g=',g)                   # Tulos
# Korvaa jokaisen stringin merkit uusilla merkeilla =====
j='xku_xnkxx' ; k=j.replace('x','a')      # Korvaa
print('j_=',j,'k=',k)                   # Tulos
# Poistetaan stringista toivotut merkit =====
l='ABNyt_OKABAB' ; m=l.strip('AB')       # Poistetaan 'AB'
print('l_=',l,'m=',m)                   # Tulos
```

► Tulostaa

```
b = 0
d = AbCd ,e= ABCD
f = EfGh ,g= efgh
j = xku xnkxx ,k= aku ankka
l = ABNyt OKABAB ,m= Nyt OK
```

String manipulointi

```
# Kommenttirivi: Tama on octave ohjelmani Omalli7.m
# – Joitakin esimerkkeja stringien manipuloinnista
#
clear ; clc
# Poistetaan ... Tyhjennetaan ...
a="karhunkierros" ; b=strfind(a,"karhu"); # Etsi teksti
printf ("b=%f\n",b)                      # Tulos
d="AbCd" ; e=toupper(d);                  # Isoiksi kirjaimiksi
printf ("d=%s,e=%s\n",d,e)               # Tulos
f='EfGh' ; g=tolower(f);                  # Pieniksi kirjaimiksi
printf ("f=%s,g=%s\n",f,g)               # Tulos
# Korvaa jokaisen stringin merkit uusilla merkeilla =====
j='xku_xnkkx' ; k=strrep (j,"x","a");     # Korvaa
printf ("j=%s,k=%s\n",j,k)               # Tulos
# Poistetaan merkkijonosta toivotut merkit.
l='ABNyt_OKABAB' ; m=strrep (l,"AB",""); # Poistetaan "AB"
printf ("l=%s,m=%s\n",l,m)               # Tulos
```

► Tulostaa

```
b= 1.000000
d= AbCd,e= ABCD
f= EfGh,g= efgh
j= xku xnkkx,k= aku ankka
l= ABNyt OKABAB,m= Nyt OK
```

Formatointi

python

- ▶ Reaaliluvun muunnos stringiksi

```
a=9.89 ; b=str(a) ; print(b)
```

tulostaa 9.89

```
print(type(a)) ; print(type(b))
```

tulostaa

```
<class 'float'> <class 'str'>
```

- ▶ Reaaliluvun formatointi stringiksi

```
a=4.567 ; b='%7.5f' %a ; print(b)
```

tulostaa 4.56700

- ▶ Reaali- ja kokonaisluku stringiksi

```
a=1.23; b=78; c='%3.1f%3i'% (a,b)
```

print(c) tulostaa 1.2 78

- ▶ Mahdotonta muistaa ulkoa vuoden (päivän, tunnin?) päästä

- ▶ **grep %i *.py** ⇒ Malli!

octave

- ▶ Reaaliluvun muunnos stringiksi

```
a=9.89 ; b=num2str(a) ; printf("%s\n",b)
```

tulostaa 9.89

```
printf("class(a)=%s\n",class(a))
```

tulostaa **class(a)=double**

```
printf("class(b)=%s\n",class(b))
```

tulostaa **class(a)=char**

- ▶ Reaaliluvun formatointi stringiksi

```
a=4.567 ; b=sprintf("%7.5f",a);
```

```
printf("%s\n",b) tulostaa 4.56700
```

- ▶ Reaali- ja kokonaisluku stringiksi

```
a=1.23 ; b=78 ; c=sprintf("%3.1f%3i",a,b)
```

```
printf("%s\n",c) tulostaa 1.2 78
```

- ▶ Lähes mahdotonta muistaa ...

grep f% *.m ⇒ Malli!

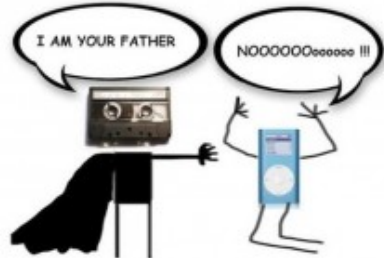
Formatointi

python (kuva:@improgrammer.net)

```
# Kommenttirivi: Tama on python ohjelmani Pmalli8.py
# — Joitakin formatointi esimerkkeja
#
import os ; os.system('clear') # Tyhjennetaan näyttö
# Reaaliluvun muunnos stringiksi =====
a=1.458 # Luku
b=str(a)
# Muunnos stringiksi
print('b=',b,',_type(b)=' ,type(b)) # Tarkistus
# Reaaliluvun formatointi: leveys=8, desimaalit=5 =====
c=7.89 # Luku
d='%8.5f' %c # Formatointi
print('d=',d) # Tarkistus
# Kokonaisluvun formatointi: leveys=6 =====
e=71 # Luku
f='%6i' %e # Formatointi
print('f=',f) # Tarkistus
# Reaaliluvun ja kokonaisluvun yhdistetty formatointi =====
g=2.34 ; h=-912 # Luvut
i='%4.1f%5i' %(g,h) # Formatointi
print('i=',i) # Tarkistus
# Reaaliluku, string ja kokonaisluku yhdistetty =====
j=9.84 ; k='tekstia' ; l=812 # Yhdistettävät
m='%6.0f%10s%i' %(j,k,l) # Formatointi
print('m=',m) # Tarkistus
# Reaaliluku, string, kokonaisluku ja ' & ' yhdistetty ==
n=9.23 ; o=-87 ; p='\\\\\\' # Yhdistettävät
q='%8.3e_&_3i_2s' %(n,o,p) # Formatointi
print('q=',q) # Tarkistus
```

► Tulostaa

```
b= 1.458 , type(b)= <class 'str'>
d= 7.89000
f= 71
i= 2.3 -912
m= 10 tekstia 812
q= 9.230e+00 & -87 \\
```



Formatointi

octave (kuva: LoadingArtist.com)

```

# Kommenttirivi: Tama on octave ohjelmani Omalli8.m
# — Joitakin formatointi esimerkkeja
#
clear ; clc                # Poistetaan ... Tyhjennetaan ...
# Reaaliluvun muunnos stringiksi =====
a=1.468;                    # Luku
b=num2str(a);               # Muunnos
printf("b_=%s_,_class(b)=%s\n",b,class(b)) # Tarkistus
# Reaaliluvun formatointi: leveys=8, desimaalit=5 =====
c=7.89;                     # Luku
d=sprintf("%8.5f",c);       # Formatointi
printf("d_=%s\n",d)        # Tarkistus
# Kokonaisluvun formatointi: leveys=6 =====
e=71;                        # Luku
f=sprintf("%6i",e);         # Formatointi
printf("f_=%s\n",f)        # Tarkistus
# Reaaliluvun ja kokonaisluvun yhdistetty formatointi =====
g=2.34; h=-912;              # Luvut
i=sprintf("%4.1f%5i",g,h);  # Formatointi
printf("i_=%s\n",i)        # Tarkistus
# Reaaliluku, string ja kokonaisluku yhdistetty =====
j=9.84 ; k='tekstia' ; l=812; # Yhdistettavat
m=sprintf("%6.0f%10s%i",j,k,l); # Formatointi
printf("m_=%s\n",m)        # Tarkistus
# Reaaliluku, string, kokonaisluku ja ' \& ' yhdistetty ==
n=9.23 ; o=-87 ; p='\\';    # Yhdistettavat
q=sprintf("%8.3e_&%3i_2s",n,o,p); # Formatointi
printf("q_=%s\n",q)        # Tarkistus

```

► Tulostaa

```
b = 1.468 , class(b)= char
d = 7.89000
f = 71
i = 2.3 -912
m = 10 tekstia 812
q = 9.230e+00 & -87 \\\
```



► **Formatointi:**

Datan lukeminen, kirjoittaminen ja “näyttäminen”

Järjestelmällisyys ja tallennettavuus

Tärkeintä on lopputulos

Sama lopputulos monilla eri tavoilla

► Mallit: “Yritys ja erehdys” n. 30 tuntia

► Lukeminen

Luetaan tiedosto **OPmalli9.dat**

Pmalli9.py vastaa $\sim$ **Omalli9.m**

Pmalli9A.py (readlines=rivilista)

Pmali9B.py (numpy.loadtxt)

Omalli9A.m (fscanf)

- Kirjoittaminen

Pmalli10.py vastaa $\sim$ Omalli10.m

Pmali11.py (numpy.savetxt)

Omalli11.m (dlmwrite)

► Näistä on teille (toivottavasti) myöhemmin hyötyä



Datan lukeminen

- **OPmalli9.dat** muotoa (Tulostus: Toinen näyttö!)

OPmalli9.dat

```
HJD      Year M   D   H   M
2454395.96 2007 10 22 10 59
2454399.48 2007 10 25 23 35
2454403.01 2007 10 29 12 11
```

Komenttirivi: Tama on python ohjelmani Pmalli9.py

```
#
import os ; os.system('clear')           # Tyhjennetaan naytto
date=[]                                  # Tyhja muuttuja
file = open('OPmalli9.dat', 'r')         # Avaa OPmalli9.dat
rivi = file.readline()                  # LUETAAN RIVI KERRALLAAN
while (len(rivi) > 0):                   # Ei tyhja rivi
    osat=rivi.split()                   # Tyhjien mukaan osiksi
    osa=osat[0]                          # 1 alkio
    q=osa.find("245")                    # Etsi "245" tekstia
    if (q>-1):                           # Jos loytyy ...,
# String muutetaan desimaaliluvuksi kaskylla "float"
        date.append(float(osa))          # Lisaa arvo muuttujaan date
    print(date)                          # Nayta nykyinen date
    rivi = file.readline()               # Lue seuraava rivi
file.close()                             # Sulje OPmalli9.dat
print("date_=",date)                     # Tarkistus
print("type(date)=",type(date))          # Tarkistus
```

Datan lukeminen

- **OPmalli9.dat** muotoa (Tulostus: Toinen näyttö!)

OPmalli9.dat

```
HJD          Year M  D  H  M
2454395.96  2007 10 22 10 59
2454399.48  2007 10 25 23 35
2454403.01  2007 10 29 12 11
```

Kommenttirivi: Tama on octave ohjelmani Omalli9.m

```
#
clear ; clc                                # Poistetaan ... Tyhjennetaan ...
date=[] ;                                  # Tyhja muuttuja
file=fopen("OPmalli9.dat","r");            # Avaa OPmalli9.dat
rivi=fgets( file );                       # LUETAAN RIVI KERRALLAAN
while ischar( rivi)==1                    # Onko string?
    osat=strsplit( rivi , '_' );           # Tyhjien mukaan osiksi
    osat=strjoin( osat(1),' ');            # 1 alkio sopivaan muotoon
    q=index( osat , "245" );               # Etsi "245" tekstia
    if ( q > 0 )                           # Jos löytyy ..,
        # String muutetaan desimaaliluvuksi kaskylla "float"
        date=[date , str2double( osat )]; # lisaa arvo muuttujaan date
    endif                                  # if loppuu
disp( date )                              # Nayta date
rivi=fgets( file );                       # Lue seuraava rivi
endwhile                                  # while loppuu
fclose ( file );                          # Sulje OPmalli9.dat
printf( "date_=%f,%f,%f\n" , date )        # Tarkistus
printf( "class( date)=%s\n" , class( date ) ) # Tarkistus
```

Datan lukeminen

- **OPmalli9.dat** muotoa (**Tulostus:** Toinen näyttö!)

OPmalli9.dat

```
HJD      Year M   D   H   M
2454395.96 2007 10 22 10 59
2454399.48 2007 10 25 23 35
2454403.01 2007 10 29 12 11
```

Kommenttirivi: Tama on python ohjelmani Pmali9A.py

#

```
import os ; os.system('clear')
f = open('OPmali9.dat', 'r')
data = f.readlines()
f.close()
date = []
for rivi in data:
    osat=rivi.split()
    osa=osat[0]
    q=osa.find("245")
    if (q>-1):
# String muutetaan desimaaliluvuksi
        date.append(float(osa))
    print(date)
print("date_=",date)
print("type(date)=",type(date))
```

```
# Tyhjennetaan naytto
# Avaa OPmali9.dat
# LUETAAN RIVILISTANA
# Sulje OPmali9.dat
# Luo tyhja lista
# Looppi datan riveille
# Tyhjien mukaan osiksi
# 1 alkio
# Etsi "245" tekstia
# Jos loytyy ...,
# kaskylla "float"
# Lisaa arvo muuttuukaan date
# Nayta nykyinen date
# Tarkistus
# Tarkistus
```

Datan lukeminen

► OPmalli9.dat muotoa

OPmalli9.dat

```
HJD      Year M  D  H  M
2454395.96 2007 10 22 10 59
2454399.48 2007 10 25 23 35
2454403.01 2007 10 29 12 11
```

```
#
# Kommenttirivi: tama on python ohjelmani Pmalli9B.py
#
import os ; os.system('clear')
# Tyhjennetaan näyttö
import numpy as np                # numpy importoitu
file='OPmalli9.dat'              # Tiedoston nimi
# Komennolla "skiprows=2" jatetaan lukematta 2 ensimmäistä rivia
date=np.loadtxt(file, skiprows=2, usecols=(0,)) # Lue 1. kolumni
print("date_=", date)             # Tarkistus
print("type(date)=", type(date))  # Tarkistus
print(np.cos(date))              # Tarkistus
```

► Tulostus

```
date = [ 2454395.96  2454399.48  2454403.01]
type(date)= <class 'numpy.ndarray'>
[ 0.00415457  0.36557712 -0.69084715]
```

► date on numpy vektori, josta voi ottaa np.cos(date)

► Käytännöllinen tulevissa laskuharjoituksissa

Datan lukeminen

► OPmalli9.dat muotoa

OPmalli9.dat

```
HJD      Year M   D   H   M
2454395.96 2007 10 22 10 59
2454399.48 2007 10 25 23 35
2454403.01 2007 10 29 12 11
```

Kommenttirivi: Tama on octave ohjelmani Omalli9A.m

clear ; clc *# Poistetaan ... Tyhjennetaan ...*

filename = 'OPmalli9.dat';

fid=**fopen**(filename, 'r');

[x,nx]=**fscanf**(fid, '%s', [1,1]);

[y,ny]=**fscanf**(fid, '%s/%s/%s/%s/%s', [6,1]);

[z,nz]=**fscanf**(fid, '%g/%g/%g/%g/%g', [6,Inf]);

fclose(fid);

printf("x_=%s,nx_=%f\n",x,nx) *# Tarkistus*

printf("y_=%s,ny_=%f\n",y,ny) *# Tarkistus*

date=z(1,:); *# 1 Sarake*

printf("date_=%f,%f,%f\n",**date**) *# Tarkistus*

printf("class(date)=_s\n",class(**date**)) *# Tarkistus*

► Tulostus

x = OPmalli9.dat,nx = 1.000000

y = HJDYearMDHM,ny = 6.000000

date = 2454395.960000,2454399.480000,2454403.010000

class(date)= double

Datan kirjoittaminen (kuva: @quotesgram.com)

```
# Kommenttirivi: Tama on python ohjelmani Pmalli10.py
# — Kirjoitetaan tiedostoon rivi kerrallaan
#
import os ; os.system('clear') # Tyhjennetaan näyttö
import numpy # Importoidaan Numpy
x=numpy.arange(3.) ; print("x=",x) # Luodaan 1 vektori
y=numpy.sin(x) ; print("y=",y) # Luodaan 2 vektori
#
# — Kirjoitetaan vain lukuja
f=open('Pmalli10a.dat','w') # Avaa tiedoston
for i in range(len(x)): # for-looppi
    f.write("%6.1f%8.4f\n" % (x[i],y[i])) # Kirjoittaa rivin
f.close() # Sulkee tiedoston
#
# — Kirjoitetaan lukuja ja stringeja esim: Latex käyttöön
f=open('Pmalli10b.dat','w') # Avaa tiedoston
for i in range(len(x)): # for-looppi
    f.write("%6.1f%3s%8.4f%4s\n"%(x[i], '&',y[i], '\\\\'))
f.close() # Sulkee tiedoston
```

© Randy Glasbergen
www.glasbergen.com



"I want a computer that does what I want it to do, not what I tell it to do!"

- ```

▶ Pmalli10a.dat muotoa
0.0 0.0000
1.0 0.8415
2.0 0.9093

▶ Pmalli10b.dat muotoa (Käytännöllinen tulevissa laskuharjoituksissa)
0.0 & 0.0000 \
1.0 & 0.8415 \
2.0 & 0.9093 \

▶ Tulostus
x= [0. 1. 2.]
y= [0. 0.84147098 0.90929743]

```

# Datan kirjoittaminen (kuva: @www.cartoonstock.com)

```
Kommenttirivi: Tama on octave ohjelmani Omalli10.m
- Kirjoitetaan tiedostoon rivi kerrallaan
#
clear ; clc # Poistetaan ... Tyhjennetaan ...
x=[0:2] ; printf("x_=%f,%f,%f\n",x) # Luodaan 1 vektori
y=sin(x) ; printf("y_=%f,%f,%f\n",y) # Luodaan 2 vektori
#
- Kirjoitetaan vain lukuja
fid=fopen('Omalli10a.dat','w'); # Avaa tiedoston
for i=1:length(x) # for alkaa
 fprintf(fid, "%6.1f%8.4f\n",x(i),y(i)); # Kirjoittaa Rivin
endfor # for loppuu
fclose(fid); # Sulkee tiedoston
#
- Kirjoitetaan lukuja ja stringeja esim: Latex kayttoon
fid=fopen('Omalli10b.dat','w'); # Avaa tiedoston
for i=1:length(x) # for alkaa
 fprintf(fid, "%6.1f%3s%8.4f%4s\n",x(i),'&',y(i),'\\');
endfor # for loppuu
fclose(fid); # Sulkee tiedoston
```

## ► Omalli10a.dat muotoa

```
0.0 0.0000
1.0 0.8415
2.0 0.9093
```

## ► Omalli10b.dat muotoa (Käytännöllinen tulevissa laskuharjoituksissa)

```
0.0 & 0.0000 \\
1.0 & 0.8415 \\
2.0 & 0.9093 \\
```

## ► Tulostus

```
x = 0.000000,1.000000,2.000000
y = 0.000000,0.841471,0.909297
```



Young Pups In A Chat Room

# Datan kirjoittaminen

```
Kommenttirivi: Tämä on python ohjelmani Pmalli11.py
– Eras nopea tapa kirjoittaa kerralla kaikki data
#
import os ; os.system('clear') # Tyhjennetaan näyttö
import numpy as np # Importoidaan Numpy
x=np.arange(3.) ; print("x=",x) # Luodaan 1 vektori
y=np.sin(x) ; print("y=",y) # Luodaan 2 vektori
#
np.savetxt('Pmalli11a.dat',(x,y),fmt='%10.3f')
np.savetxt('Pmalli11b.dat',np.transpose((x,y)),fmt='%10.3f')
```

## ► Pmalli11a.dat muotoa

```
0.000 1.000 2.000
0.000 0.841 0.909
```

## ► Pmalli11b.dat muotoa

```
0.000 0.000
1.000 0.841
2.000 0.909
```

## ► Tulostus

```
x= [0. 1. 2.]
y= [0. 0.84147098 0.90929743]
```

## ► Nopea tapa tallentaa: yhden rivin kommento!

# Datan kirjoittaminen

```
Kommenttirivi: Tama on octave ohjelmani Omalli11.m
– Eras nopea tapa kirjoittaa kerralla kaikki data

clear ; clc # Poistetaan ... Tyhjennetaan ...
x=[0:2] ; printf("x_=%f,%f,%f\n",x) # Luodaan 1 vektori
y=sin(x) ; printf("y_=%f,%f,%f\n",y) # Luodaan 2 vektori
a=[x;y]; # Data matriisiin

dlmwrite('Omalli11a.dat',a,'delimiter','\t','precision','%10.3f')
dlmwrite('Omalli11b.dat',a,'delimiter','\t','precision','%10.3f')
```

## ► Omalli11a.dat muotoa

|       |       |       |
|-------|-------|-------|
| 0.000 | 1.000 | 2.000 |
| 0.000 | 0.841 | 0.909 |

## ► Omalli11b.dat muotoa

|       |       |
|-------|-------|
| 0.000 | 0.000 |
| 1.000 | 0.841 |
| 2.000 | 0.909 |

## ► Tulostus

```
x = 0.000000,1.000000,2.000000
y = 0.000000,0.841471,0.909297
```

## ► Nopea tapa tallentaa: jälleen vain yhden rivin komento!

## ► \t = “Represents a horizontal tab” Löytyy esimerkiksi täältä | [www.cplusplus.com/reference/string/string/t/](http://www.cplusplus.com/reference/string/string/t/)