

Gruppe 1 Lösungen Einarbeitung

Aufgabe 1

src: <https://www.redhat.com/sysadmin/jobs-bg-fg>

Um einen Befehl von Anfang an im Hintergrund auszuführen, ein `&` anfügen

- `sleep 500 &`

Um einen Befehl nach dem Ausführen in den Hintergrund und zurück zu bewegen:

- `sleep 500`
- Während der Ausführung: `control + Z` um den aktuell laufenden Befehl zu pausieren.
- Mit `jobs` die Liste der aktuell laufenden Prozesse auflisten (auch mit diesen Flags: `-l`, `-n`, `-p`, `-r`, `-s`)
- Mit `bg %x` den Befehl in den Hintergrund schieben und fortsetzen, wobei `x` die Jobnumber ist (möglich sind auch: `%abc`, `!?abc`, `%-`)
- Mit `fg %x` den Befehl in den Vordergrund hohlen, wobei `x` die Jobnumber ist (möglich sind auch: `%abc`, `!?abc`, `%-`)

Aufgabe 2

```
from datetime import datetime
import time

class HelloWorld():

    def printHello(self):
        print("Hello")
    def printALot(self):
        for i in set("Welcome to my TED talk!"):
            print(i)
            time.sleep(0.5)
    def waitATon(self, timer):
        t1 = 0
        # src: https://stackoverflow.com/questions/28154066/how-to-convert-
        datetime-to-integer-in-python
        t2 = int(datetime.now().strftime('%Y%m%d%H%M%S')) + timer
        while t1 < t2:
            t1 = int(datetime.now().strftime('%Y%m%d%H%M%S'))
        print("loop finished")

a = HelloWorld()
a.printHello()
a.printALot()
a.waitATon(25)
```

Augabe 3

src: <https://linuxhandbook.com/uid-linux/>

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main(){
    ifstream fichier ("/etc/passwd");
    string tab[266], token[228];

    if(fichier){
        string ligne;
        int i=0, compt=0, comptligne=0;
        cout<<"affichage ligne par ligne"<<endl;
        while(getline(fichier, ligne)){
            tab[i]=ligne;
            i++;
            comptligne++;
        }

        string delim=":";
        size_t pos=0;

        for (i = 0; i < comptligne; i++)
        {
            while ((pos=tab[i].find(delim)) != string::npos)
            {
                token[compt] = tab[i].substr(0, pos);
                tab[i].erase(0, pos+delim.length());
                compt++;
            }
        }

        for (i = 0; i < size(token); i+=6)
        {
            cout<<"User-Namen : "<<token[i]<<"    User-Id (UID) : "
<<token[i+2]<<endl;
        }
    }
    else
        cout<<"Impossible d'ouvrir le fichier"<<endl;

    return 0;
}
```

Aufgabe 4

- Abstraktion von Hardware für Programme
- Verwaltung von Ressourcen
- Schutz der Hardware vor direkten Zugriffen
- Zulassen und Abgrenzung mehrerer Anwender
- Virtualisierung des Speichers

Abgaben von Anton Stimmer, Oscar Röth