

Linux 1 – Inlämningsuppgift 2

Den här inlämningsuppgiften ska utföras på **13.53.57.72**, precis som den förra.

Målet med uppgiften är att skapa två skript, "pdftosite" och "servepdf".

Skriptet "pdftosite" konverterar en pdf-fil till en liten webbsite, en samling med html-filer och bilder i en mappstruktur. De pdf-filer som skriptet ska klara av är de presentationer vi använder i kursen.

Skriptet "servepdf" använder det första skriptet för att skapa site-filer i en temporär mapp och startar sen en webbrowser som servar filerna därifrån.

Setup

Webbserver

Vi kommer använda oss av en superenkel webbrowser som kommer med Python 3:

```
$ python3 -mhttp.server
```

Raden ovan startar en webbrowser på port 8000 och servar filer från nuvarande katalog.

Men – eftersom vi alla delar samma server så måste alla köra sin webbrowser på en unik port.

Välj port (uid + 8000). Ditt uid får du om du skriver kommandot id. Om du, av någon anledning, behöver ytterligare en port adderar du 100 till din första. Användaren miklar får därför port 9032 med alternativ port 9132, 9232 osv. På detta sätt kommer vi inte att fajtas över portar.

User miklar ska använda port 9032 och startar servern så här:

```
$ python3 -mhttp.server 9032
```

ssh med tunnel

För att enklast kunna testa och utveckla våra skript sätter vi upp en ssh-tunnel. Det kan vi göra i samband med att vi ansluter till servern med ssh:

```
ssh -L 8080:localhost:9032 <user>@13.53.57.72
```

Raden ovan kopplar upp och loggar in dig, precis som vanligt, men sätter också upp en tunnel så att du kan nå din server på serverns port 9032 via din lokala dators port 8080. S.k. port forwarding

Du anger så klart ditt portnummer som du kör webservern på istället för 9032!

Om du har tunneln uppe och webbrowsern snurrande igen, testa att öppna "localhost:8080" i din webbläsare. Du borde nu se en listning av innehållet i mappen där du startade webbrowsern.

Du kan så klart välja en annan lokal port än 8080. Text samma som du använder på servern så slipper du krockar med eventuell webbrowser du kör lokalt.

Skriptet "pdftosite"

- Skriptet tar två argument, <pdf> och <destinationsfolder>.
- Skriptet extraherar text och bilder från pdf:en och skriver till <destinationsfolder> i form av html-filer och jpg-bilder.
- Extraheringen skapar en html-sida per pdf-sida och skriver resultatet till <destinationsfolder>
- En bild extraheras från varje pdf-sida och sparas i <destinationsfolder>.
- Bilder som extraherats ska flyttas till <destinationsfolder>/img och konverteras till gråskala och 600 pixlars bredd. Detta för att spara plats.

Resultatet blir ett filträd med följande innehåll:

```
├─ site
│   ├── img
│   │   ├── img001.jpg
│   │   ├── img002.jpg
│   │   │
│   │   │
│   │   ├── img022.jpg
│   │   └── img023.jpg
│   ├── index-1.html
│   ├── index-10.html
│   │
│   │
│   ├── index.html
│   └── index_ind.html
```

- Skriptet ska validera inargument och vägrar skriva över existerande destinationsmapp.
- Skriptet ska meddela användaren om det anropas fel.
- Skriptet ska använda funktioner för olika delar efter behov.
- Efter att bilderna konverterats ska bara de nya filerna finnas kvar, inte originalbilderna som extraherats från pdf:en.
- Skriptet ska inte producera någon utskrift från de kommandon som används.

För att konvertera pdf till text och bilder har jag installerat paketet "poppler-utils" på servern, som bland annat innehåller verktygen pdftotext, pdftohtml och pdftimages.

För att konvertera och skala om bilderna har jag installerat paketet "netpbm", där vi kommer behöva använda verktygen:

- pngtopnm
- jpegtopnm
- pnmscale
- ppmtojpgm
- ppmtojpeg

Verktygen från "netpbm" – paketet är gjorda för att pajpas till varandra, och har man-sidor som beskriver de flaggor som kan behövas.

- Inga andra konverteringsverktyg får användas
- Skriptet ska kunna konvertera godtycklig presentation från kursmaterialet (Lektion-X.pdf)

Skriptet "servepdf"

- Skriptet ska använda "pdftosite"-skriptet från föregående steg.
- Skriptet skapar och använder en dold mapp i användarens hemkatalog för tillfällig lagring.
- I den dolda mappen skapas sedan pdf – siden med hjälp av "pdftosite"-skriptet.
- Skriptets accesslogg (från webbservern) ska hamna i en loggfil i den dolda mappen.
 - o Loggfilen skrivs över vid varje körning
- Om mappen innehåller en tidigare genererad site ska den tas bort när "servepdf" startar.
 - o Det går alltså inte att köra två "servepdf" samtidigt för en användare. Det är OK.
- Skriptet tar två argument, <pdf> och [port].
 - o pdf-argumentet är pdf:en som ska servas.
 - o port-argumentet anger porten som servern ska lyssna på. Om det inte anges väljs standardporten 8000.
- Skriptet klagar snällt om det anropas fel. Text om ingen port angivits.
- Skriptet ska meddela när konverteringen är körd och är klar, exempelvis:

```
$ ./servepdf "Lektion 2.pdf" 9032
```

```
Converting...done!
```

```
Starting server on port 9032
```

Tips

- För att skapa en mapp och alla föräldermappar på vägen kan man använda "mkdir -p"
 - o Om jag i en tom mapp exempelvis skriver "mkdir -p a/b/c/d" så skapas alla fyra mapparna.
 - o Om exempelvis redan "a/b" finns skapas bara resterande undermappar.
- För att få fram enbart filnamn från en hel sökväg kan man använda "basename:"

```
frasse@frasse:~/inlup2$ basename pdfsite/page1/index.html  
index.html
```

- Bildfilerna ska börja på img, ha indexsiffror och sluta på .jpg. De måste vara jpeg-filer.
- Resultande webbsida blir troligen ganska snygg. Du får också göra den finare om du kan/vill.
- Skripten ska ligga bredvid varandra i din arbetsmapp och det är tänkt att de ska startas med "./". De behöver alltså inte installeras i någon "bin" – katalog. När "servepdf" ska anropa "pdftosite" kan det alltså referera till "./pdftosite".
- Använd man-blad och "--help" om information i uppgiften inte är tydlig med hur ett kommando ska anropas.
- Fråga om du kör fast!

Utförande och resultat

- För över minst en av Lektions-pdf:erna till servern. Använd scp.
- Skapa en arbetsmapp i din hemkatalog.
- Skapa en symbolisk länk till din arbetsmapp i mappen `/opt/linux1/din-användare`, kalla länken `inlup2`.
- Om din användare är `frasse`, skapa alltså länken `/opt/linux1/frasse/inlup2` som pekar på din arbetsmapp.
- Testa att göra `cd /opt/linux1/din-användare/inlup2` för att se att länken är korrekt.
- Kopiera mall-skriptet för `pdftosite` från `/opt/linux1/inlup2/pdftosite.template` till din arbetsmapp och följ instruktionerna i den, om det står att du ska skriva en förklarande text till något som redan är skrivet i den så ska du göra det. (Det finns inget mall-skript för `servepdf`.)
- Så snart du har något resultat att titta på i form av extraherade bilder, använd tunneln och din webbläsare för att verifiera ditt resultat.
- När du fått konverteringen att fungera, testa med ett par andra pdf:er från kursens presentationer. Flytta alla pdf:er du testat till mappen `<user>/inlup2/pdf`.
- Din arbetsmapp ska innehålla resultatet av att konvertera `lektion2.pdf` i en undermapp som heter `lektion2`.
- När du är klar, skapa en tom fil som heter `DONE` i din arbetsmapp. Inget får ändras efter att filen `DONE` skapats.

Förväntat resultat i din arbetsmapp:

```
pdftosite
servepdf
pdf/
lektion2/
DONE
```

Max antal poäng: 30

Poäng för godkänt (60%): 18

Sista inlämningsdag: 12 April 2023