

L^AT_EX rapport til oblig 1

Arselan Sultani *

September 19, 2014

Abstract

Denne rapporten skal leveres angående obligatorisk innlevering 1 i INF3331 og den er skrevet i L^AT_EX . Denne skal leveres sammen med shell scriptene og python scriptet på Github.

1 Del 1

1.1 list_new_files.sh

Den finner de filene som er endret i n siste dager og mappen som dn skal lete i. Navnet på mappen og n får vi fra brukeren som argumenter. -mtime 2.arg: sier om antall dager -print0: er dersom der er mellomrom i navn på filen, fordi ellers vil den leste samme fil som to forskjellige navner. -B k: er for å få med K bak fil størrelsen

1.2 find_word.sh

Her skal vi lete i filer i mappen gitt, etter en ord som blir gitt av brukeren. -grep 2.argument: er ordet vi skal finne i de filene

1.3 sized_delete.sh

Her skal vi finne alle de filene som er større enn en gitt størrelse og slette de. -size 2.arg: vil si størrelsen av filen vi søker etter er større enn 2.arg -exec rm: vil si at den skal slette uten å spørre om tillatelse fordi dersom vi hadde sagt -ok rm, ville den ha funnet de filene men spurt en og en om vi ville fjerne de.

1.4 sort_file.sh

Her skal vi lese ord fra en fil og sortere de og så skrive ut til en annen fil. Begge filene blir gitt av brukeren. Og dersom output filen ikke eksisterer, vil den bli opprettet og skrevet i. -o 2.arg er output-filen

*arselans@ulrik.uio.no.

2 Oppgave 2

Den starter ved å sjekke argumenter, og hvis det er mindre enn 4, så skriver den ut en feilmelding. Hvis ikke, så skriver den ut først target, så files osv. Så kaller vi på metoden `generate_tree`.

2.1 `generate_tree()` og `mkDirs()`

Jeg har antatt at vi skulle finne random dypde fra root. Og det skal være mellom 2 og det er som gitt. Så skal den finne random tall mellom 2 til dir fordi jeg har antall at også dette skal være random. Så kaller den på `mkDirs` metoden som lager mappene. Så sier at så lenge dypde er større eller lik 0, så skal den gå videre. Så lenge x er mellom 1 og en random tall for antall mapper i mappen, lager jeg en ny path i `new_dir2`. Den tar den pathen som den allerede har og legger en `\` i mellom og så en tilfeldig string fra `random_string()` som blir navnet på den nye mappen. `os.makedirs` lager mappen. Deretter blir det sjekket om `verbose` er `True` eller `False`, for dersom den er `True`, så skal den skrive pathen til den nye mappen. Så kalles `populate_tree()`, som fyller mappene med filer. Og den vil gå i rekursjon. Den vil eventuelt bli stoppet av `for-løkka` og `dir_deep` som minsker hver gang vi lager en mappe inn i en mappe fordi da er vi lenger unna root.

2.2 `populate_tree()`

Den vil først lage en `last_modified` og `last_accessed`. `last_modified` skal være mellom `start_time` og `end_time`, som vi får fra argumentene og `last_accessed` skal være mellom `last_modified` og `end_time`, fordi jeg antar at du kan få tilgang til filen uten å ha endret på den. Deretter lages det filer i mappen. Filnavnet blir opprettet og vi spør om vi kan åpne filen. Og dersom filen eksisterer, så kan skal vi åpne den og overskrive. Men dersom filen ikke eksisterer, så oppretter vi den. Dersom `verbose` er `True`, så skriver vi ut pathen til filen som vi nettopp lagde. Deretter finner vi størrelsen til filen og fyller den etter til det blir så mye som den vil ha. Og til slutt closer vi den.