Denne oppgåva blir ikkje rekna med i sluttresultatet

Obligatorisk Innlevering 1 Shell

Til første obligatoriske innlevering skal du lage ditt eget skall (shell) lignende bash.

Vi har laget et lite program **my_shell_skeleton.c (https://mitt.uib.no/courses/42392/files/5308172?wrap=1)** ⭳ **(https://mitt.uib.no/courses/42392/files/5308172/download?download_frd=1)** du kan starte fra. Dette programmet leser én kommando fra terminalen og kjøre den. Før du starter med oppgavene under er det anbefalt å sette seg inn i denne koden og forstå hva som skjer. Les gjerne manualen til hver av funksjonene **getline**, **strtok**, **exit** og **execvp**. Kommandoen for å åpne manualen til en funksjon er **man**, for eksempel `man getline`.

Oppgavene i denne innleveringen skal løses med programmeringsspråket **C** og kjøre på **Linux**. Husk at dette er et krav, så sjekk at alt fungerer på en virtuell maskin med Linux dersom du velger å programmere lokalt på Windows eller Mac. Filene skal leveres gjennom CodeGrade, en fil per deloppgave.

Oppgave 1

Koden du har fått utdelt leser én kommando fra brukeren og kjører den. Den første oppgaven er få dette til å skje i en løkke. Programmet skal altså ikke terminere etter én kommando, men heller vente på at kommandoen fullfører, og så vente på neste input. Du trenger funksjonene **fork**, **execvp** og **wait** for å løse denne oppgaven. Du bør først lese manualen til disse tre funksjonene for å forstå hvordan de fungerer. **Kapittel 5** i boken er også et bra sted å begynne.

Når du er ferdig skal programmet fungere som dette:

Oppgave 2

På dette tidspunktet kjører programmet i en uendelig løkke og bare sender videre hver kommando. Det er ingen måte å stoppe programmet på uten å bruke `Ctrl + C` eller lignende. I

denne oppgaven skal du sjekke innputen fra brukeren, og dersom brukeren skrev **exit** skal programmet avsluttes. Det skal også være mulig for brukeren å trykke `Enter` med en tom linje uten at dette fører til segmentation fault eller andre feilmeldinger.

Her er et eksempel på hvordan en løsning skal fungere:

Oppgave 3

Det neste du skal implementere er muligheten for å omdirigere resultatet fra en kommando til en fil. Syntaksen for dette skal være

`[kommando] > [fil]`

For eksempel

`pwd > dir.txt`

For å implementere dette må du vite hvordan man endrer standard output streamen (**stdout**) til en prosess. Det er flere måter å gjøre dette på, men igjen er **kapittel 5** i boken et bra sted å begynne. Funsjoner som kan være nyttig her er **open**, **close**, **fopen**, **dup2** og **fileno**.

Husk at input fra brukeren nå er `[kommando] > [fil]`, men kun kommandoen skal videre til **execvp**.

Når du er ferdig med denne oppgaven skal dette fungere:

Oppgave 4 (valgfri)

Du har kanskje merket at når du bruker `Ctrl + C` i andre skall sendes **SIGINT** til prosessen som kjører, men skallet blir værende. I ditt program nå vil `Ctrl + C` avslutte både kommandoen som kjører og selve skallet. I denne siste oppgaven skal du lage din egen ***handler*** for **SIGINT** signalet slik at kun kommandoen som kjører stopper.

Se denne **artikkelen** (https://linuxhint.com/signal_handlers_c_programming_language/) for å komme i gang.

Når du er ferdig skal skallet ditt støtte dette:

Du må laste dette verktøyet i eit nytt nettlesarvindauge

Last Obligatorisk Innlevering 1 Shell i eit nytt vindauge