پروژه ۱ فصل ۳ _ پوسته یونیکس و امکان history

هدف :

طراحی یک رابط پوسته برای پذیرفتن فرمان های کاربر و اجرا هر کدام در پردازه ای مجزا

سیستم عامل های قابل اجرا :

لینوکس ، یونیکس ، Mac OS X

ورودی :

فرمان های کاربر با هر اعلان osh> توسط رابط پوسته

خروجی :

حالت۱ : پردازه والد خط فرمان را خوانده و پردازه فرزند مجزایی ایجاد کند که فرمان را اجرا کند.

حالت۲ : پردازه والد پیش از ادامه اجرا منتظر می ماند تا فرزند خارج شود. **کاربر می تواند با قرار دادن علامت ampersand(&) در انتها فرمان تعیین کند که پردازه های والد و فرزند اجرای همروند داشته باشند.**

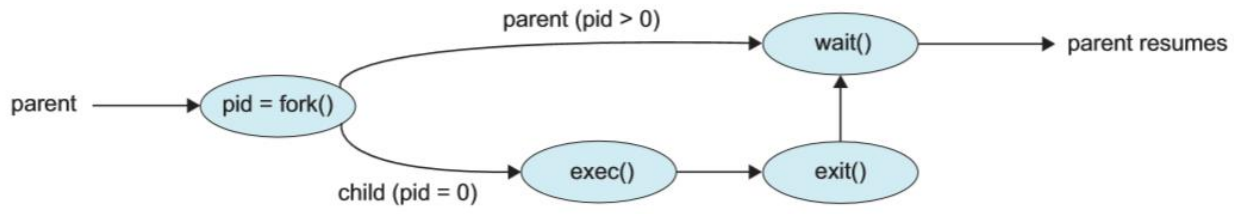

Figure 3.10 Process creation using the `fork()` system call.

قابلیت ها :

خواند ورودی کاربر و ذخیره آن ها به صورت پارامتر هایی مجزا در آرایه args توسط تابع

void getInputs(char input[], char *args[],int *ampersand)

مقادیر آرایه args نیز قابل مشاهده است.

ادامه خواندن فرمان های کاربر تا زمانی که کاربر فرمان exit را صادر کند که should_run را برار ۰ قرار می دهیم.

پردازه کاربر با اجرای تابع زیر

execvp(char *command, char *params[]);

با ورودی execvp(args[0], args) دستور کاربر را اجرا می کند.

اگر کاربر علامت ampersand را بگذارد ، مقدار متغیر ampersand را برابر ۱ قرار داده و والد تا اجرا فرزند صبر می کند.

نام فرمان های کاربر در آرایه history در هر اعلان ذخره می گردد.

کاربر قادر است مجموعه فرمان های خود را از اخیرترین به قدیمی ترین دستور مشاهده کند برای مثال :

ps, ls -l, top, cal, who, date

با اجرای دستور history به صورت :

6 ps

5 ls -l

4 top

3 cal

2 who

1 date

نمایش میگردد. (در هنگام ذخیره سازی تنها نام فرمان و نه ورودی ها آن ذخیره می گردد)

امکان ذخیره سازی تنها ۱۰ فرمان وجود دارد.

کاربر با وارد کردن علامت !! ، اخیر ترین فرمان سابقه (در مثال بالا ps) را اجرا می کند اگر سابقه ای نبود پیغام "No commands in history" چاپ می گردد.

کاربر با وارد کردن علامت ! همراه با یک عدد منفرد صحیح ، فرمان متناظر آن را از سابقه اجرا می گردد. برای مثال !۳ فرمان cal را در مثال بالا اجرا می کند. اگر سابقه ای نبود پیغام "No commands in history" چاپ می گردد.