محمدصالح پژند 400521171

تابع execute_args :

این تابع طراحی شده است تا دستورات را در یک محیط شل اجرا کند. این تابع همچنین دستورات داخلی و خارجی را پشتیبانی می‌کند. این کار با دستور execvp امکان پذیر شده است.

پارامتر args که به عنوان ورودی گرفته میشود یک اشاره‌گر به یک آرایه از رشته‌ها است که دستور و آرگومان‌های آن را نمایان می‌کند.

همانطور که در سوال گفته شده ما از دو نوع دستورات استفاده میکنیم. یکی دستورات داخلی که خودمان طراحی کرده ایم مانند cd و exit و دیگری دستورات خارجی که به صورت طبیعی به کمک execvp syscall اجرا میشوند.

Execvp در واقع یک سیستم کال است که یک برنامه را جایگزین فرآیند فعلی می‌کند. این سیستم کال برنامه مشخص‌شده را با برنامه جدید از طریق یک فایل اجرایی جایگزین می‌کند.

در مورد دستور exit اگر برای این دستور خودمان چیزی تعریف نکنیم با اجرای ان توسط execvp کل برنامه شل اصلی و ترمینال بسته میشود ولی با تعریف ان به صورت جدا تعیین میکنیم که فقط از شلی که خودمان ساخته ایم خارج شود.

در مورد `cd` نیز، این دستور نیازمند تغییر دایرکتوری در محیط شل است و تغییرات باید در همان فرآیند جاری اعمال شود. اگر از execvp برای اجرای cd استفاده کنیم، یک فرآیند جدید برای اجرای دستور cd ایجاد می‌شود. این فرآیند جدید تغییر دایرکتوری را انجام می‌دهد، اما این تغییرات تنها در این فرآیند جدید قابل مشاهده است و تغییرات در فرآیند اصلی شل که روی ان کار میکنیم تاثیری ندارد.

به همین دلیل، برای اجرای دستورات exit و cd از توابع دستور داخلی مانند exit و chdir به جای execvp استفاده می‌شود.

تابع own_cd :

این تابع به منظور کنترل دستور داخلی برای تغییر دایرکتوری کنونی در یک محیط شل است.

پارامتر args که به عنوان ورودی گرفته میشود یک اشاره‌گر به یک آرایه از رشته‌ها که دستور و آرگومان‌های آن را نمایان می‌کند. انتظار می‌رود که آرگومان دوم، دایرکتوری مقصد برای تغییر دادن دایرکتوری کنونی باشد.

در ابتدا بررسی می‌کند که آیا آرگومان مورد نیاز برای تغییر دایرکتوری فراهم شده است یا نه. اگر نه، یک پیام خطا به `stderr` چاپ می‌شود.

اگر از مرحله قبل گذر کرد یعنی چنین فایلی وجود دارد و از chdir برای تغییر دایرکتوری کنونی به دایرکتوری مقصد استفاده می‌کند.