

Variable de entorno- Una variable de entorno es una variable externa a un programa o proceso informático que tiene un impacto en su comportamiento.

Las variables de entorno son utilizadas por el sistema operativo y las aplicaciones para determinar información sobre el entorno de ejecución y personalizar el comportamiento de los programas en consecuencia.

Esta función parece ser una implementación de una shell en C. Aquí está una explicación detallada de lo que hace:

1. Obteniendo las variables de entorno: La función `BuscarVariable` se utiliza para buscar variables de entorno en `args[i]`. Si se encuentra una variable de entorno, se añade a `new_envp[i]`.
2. Creando el proceso: Se utiliza la función `fork()` para crear un nuevo proceso. Si `fork()` devuelve -1, se informa de un error y se sale de la función.
3. Proceso hijo: Si el PID es 0, entonces estamos en el proceso hijo. Aquí, se parsean los argumentos para determinar si el programa debe ejecutarse en segundo plano y cuál es su prioridad. Luego, se ejecuta el programa utilizando `execvp` o `execvpe`, dependiendo de si se encontraron variables de entorno.
4. Proceso padre: Si el PID es mayor que 0, entonces estamos en el proceso padre. Aquí, se comprueba si el último argumento es '&', lo que indica que el programa debe ejecutarse en segundo plano. Si es así, se crea un nuevo elemento de proceso y se añade a la lista de procesos. Si el último argumento no es '&', entonces se espera a que el proceso hijo termine utilizando `waitpid`.
5. Finalmente, la función devuelve `SSUCC_EXIT` si el código de salida del proceso hijo es `SCSS_EXIT`, y `FSUCC_EXIT` en caso contrario.

- `extern char **environ;` es una declaración en C que se utiliza para acceder a las variables de entorno del sistema. Aquí está una explicación más detallada:
- `environ` es un puntero a un array de punteros a cadenas de caracteres, donde cada cadena representa una variable de entorno.
- Cada variable de entorno tiene la forma "nombre=valor". El nombre es sensible a mayúsculas y minúsculas y no puede contener el carácter '='. El valor puede ser cualquier cosa que pueda representarse como una cadena.
- El último puntero en este array tiene el valor `NULL`.
- Este array de cadenas se pone a disposición del proceso cuando se inicia un nuevo programa mediante la llamada a `execve(2)`.
- Cuando se crea un proceso hijo mediante `fork(2)`, se hereda una copia del entorno de su proceso padre.
- Un programa en C puede manipular su entorno utilizando las funciones `getenv(3)`, `putenv(3)`, `setenv(3)` y `unsetenv(3)`.