SISTEMAS OPERATIVOS II

Tercer curso Ingeniería Técnica Informática de sistemas. Curso 2007-2008

Práctica 3: Procesos en UNIX: Entorno

Continuar la codificación de un intérprete de comandos (shell) en UNIX. Al igual que en la práctica anterior

- los argumentos entre corchetes [] son opcionales.
- Los argumentos separados por `|` indican que debe ir uno u otro, pero no ambos simultaneamente.
- Cuando el shell no pueda ejecutar una acción por algún motivo, debe indicarlo con un mensaje como el que se obtiene con `sys_errlist[errno]` o con *perror()*
- El shell no debe dilapidar memoria. La memoria que se asigna (con *malloc()*) y deja de utilizarse debe ser liberada.
- La información que se nuestra en pantalla no debe incluir en ningun caso líneas en blanco.
- El shell leerá de su entrada estándar y escribirá en su salida estándar, de manera que podría ser ejecutado un archivo de comandos invocando el shell con su entrada estándar redireccionada a dicho archivo.

entornomain Muestra la lista de todas las variables de entorno accediendo mediante el tercer argumento de *main()*. Para cada variable de entorno muestra, en una misma línea, la variable (nombre=valor), la dirección de memoria donde se almacena la variable (el valor de la variable e[i] como puntero) y la dirección de memoria donde se almacena el puntero (&e[i]).

entornoenviron Igual que el anterior, pero el acceso es mediante la variable em environ

entornos Muestra el valor (como punteros) del tercer argumento de main y de *environ* así como las direcciones de memoria donde están almacenados el tercer argumento de main y environ. Ejemplo

```
#entornomain
......
.......
.......
ffbefcb8->env[29]=(ffbeff86) PATH=/usr/bin:/usr/local/bin:
```

```
ffbefcbc->env[30]=(ffbeffa5) LC_MONETARY=es
ffbefcc0->env[31]=(ffbeffb4) LC_COLLATE=es
ffbefcc4->env[32]=(ffbeffc2) _INIT_NET_STRATEGY=none
#entornos
20a34->environ=ffbefc44 | ffbefc24->env=ffbefc44  | ffbefc20->argv=ffbefc3c
#
```

get var1, var2 ... Devuelve los valores de las variables de entorno var1, var2 .. obtenidos mediante el tercer argumento de *main*, *environ* y la función de librería *getenv*. Para cada caso imprime además la dirección de memoria donde está la variable (su valor como puntero) y la dirección donde se almacena el puntero (salvo en el caso de obtenerla con *getenv*).

put -e|-a|-p var valor Establece el valor de la variable de entorno *var* a *valor* accediendo a ella mediante *environ* (-e), el tercer argumento de main (-a) o la función de librería *putenv*. Si la variable no existe no la creará (informará de ello), salvo que se haya especificado que se haga mediante *puntenv*, entonces será creada.

sus -e|-a var nue=val Sustituye de la variable de entorno *var* por *nue=val* accediendo a ella mediante *environ* (-e) o el tercer argumento de main (-a). Si la variable *var* no existe informará de ello.

- modificar los comandos **xec, xecpri, pri** y la ejecución en primer y segundo plano (*com* y *com &* para que pueda un entorno alternativo. El formato es
  [xec|xecpri|pri] [lista_variables] ej [args] [&]
  donde
  - Si no se especifica lista_variables se entenderá que la ejecución es mediante *execv*. Lista variables puede contener solamente nombres de variables (en este caso su valor se obtiene de *environ*) o cadenas de la forma variable=valor (en este caso la variable no tiene por que existir previamente). El programa se ejecutará con un entorno que contiene UNICAMENTE las variables de la lista, usando para ello *execve*. Si como lista indicamos la cadena 'NULL' el programa se ejecutará con un entorno vacio.
  - Como en la práctica anterior, al ejecutable podrá pasársele un número variable de argumentos y ademaás debe residir en uno de los directorios de la ruta de búsqueda o especificarse la trayectoria hasta él comenzando por /, ./ o ../

  ejemplos

```
# xterm -e ksh &

# xec TERM DISPLAY HOME HZ=120 N=65 ./a.out arg1 arg2

# TERM DISPLAY HOME USER NUEVA=PRUEBA /usr/bin/xterm -hold -e ./a.out

# NULL ./a.out
```

Las variables no tienen por que ir en mayúsculas, la primera cadena que no sea una variable (y se sabe que es una variable si existe en *environ* o es una cadena en la forma cad1=cad2) es el ejecutable

Información detallada de las llamadas al sistema y las funciones de la librería debe obternerse con man (execve, wait4, getenv, putenv..)

FORMA DE ENTREGA Como en prácticas anteriores

FECHA DE ENTREGA VIERNES 18 ENERO 2008