Boletín 2: Redirección, *prompt* y comandos internos

Ampliación de Sistemas Operativos

Dpto. Ingeniería y Tecnología de Computadores (DITEC)

Universidad de Murcia

Curso 2017/2018

Índice

1. Redirecciones en `simplesh`

Redirecciones en simplesh

- Cuando se redirige la salida a un fichero, sus permisos no permiten leer su contenido:

```
simplesh> echo linea1 > listado
simplesh> cat listado
cat: listado: Permiso denegado
simplesh> chmod u+r listado
simplesh> cat listado
linea1
```

- Además, el comportamiento de >> no es el esperado:

```
simplesh> echo linea1 >  listado
simplesh> echo linea2 >> listado
simplesh> chmod u+r listado
simplesh> cat listado
linea2
```

Redirecciones en `simplesh`

- (0.25p) EJERCICIO 1: Modifica `simplesh` para que los ficheros se creen con permisos 700 y la semántica de las redirecciones sea la misma que tendrían si se ejecutasen los comandos en `bash`
- Para realizar el ejercicio, utiliza los *flags* y el *mode* de `open()`

2. *Prompt* en `simplesh`

El símbolo del sistema o *prompt* de `simplesh`

- La llamada `readline` muestra el *prompt* y, a continuación, devuelve la línea de comandos introducida por el usuario

```
1  char* get_cmd()
   {
3      char* buf= readline("simplesh> ");

5      if (buf)
           add_history(buf);
7
       return buf;
9  }
```

- Por ejemplo:

```
1  simplesh> ls > listado # ‘buf‘ apunta a "ls > listado"
```

El símbolo del sistema o *prompt* de `simplesh`

- (0.25p) EJERCICIO 2: Modifica el *prompt* para que muestre el usuario y el directorio de trabajo actual separados por el caracter `@`. Por ejemplo:

```
1  usuario@directorio>
```

- Para ello, únicamente se pueden usar llamadas POSIX o funciones de la librería estándar de C (C11)
- Llamadas `POSIX` y/o `glibc` a tener en cuenta:
  - `getuid()`
  - `getpwuid()`
  - `getcwd()` (Versión `POSIX` sin extensiones `GNU`)
  - `basename()` (Versión `POSIX`)

3. Comandos internos

El comando interno `cwd`

- (0.5p) EJERCICIO 3: Implementa el comando interno `cwd`:

```
simplesh> cwd
simplesh: cwd: /ruta/a/dir
simplesh> cwd | tr / : | cut -d : -f 4
dir
```

- `simplesh: cwd:` debe enviarse a `stderr` y `/ruta/a/dir` a `stdout`
- Para ello, únicamente se pueden usar llamadas al sistema POSIX o funciones de la librería estándar de C (C11)
- Llamadas `POSIX` y/o `glibc` a tener en cuenta:
  - `getcwd()` (Versión `POSIX` sin extensiones `GNU`)
- Para implementar `cwd`, escribe la función `run_cwd()` e inserta llamadas a la misma en `simplesh.c` donde sea necesario

El comando interno `exit`

- (0.5p) EJERCICIO 4: Implementa el comando interno `exit`:

```
simplesh> exit
[bash]
```

- Para ello, únicamente se pueden usar llamadas al sistema POSIX o funciones de la librería estándar de C (C11)
- Llamadas `POSIX` y/o `glibc` a tener en cuenta: `exit`
- Para implementar `exit`, escribe la función `run_exit()` e inserta llamadas a la misma en `simplesh.c` donde sea necesario
- Nótese que `exit` sólo termina el proceso que ejecuta la llamada
- Verifica que `simplesh` finaliza siempre que encuentra el comando interno `exit` (el segundo `echo` no se ejecuta)

```
simplesh> echo 1 ; exit ; echo 2
1
[bash]
```

- Comprueba que tras ejecutar `exit` se libera toda la memoria

El comando interno cd

- (0,5p) EJERCICIO 5: Implementa el comando interno `cd dir`:

```
simplesh> cwd
simplesh: cwd: /home/usuario
simplesh> cd directorio
simplesh> cwd
simplesh: cwd: /home/usuario/directorio
```

- Para ello, únicamente se pueden usar llamadas al sistema POSIX o funciones de la librería estándar de C (C11)
- Llamadas `POSIX` y/o `glibc` a tener en cuenta: `chdir()`
- Nótese que `chdir()` sólo cambia el directorio actual del proceso que ejecuta la llamada
- Verifica que `simplesh` modifica el directorio actual siempre que encuentra el comando interno `cd`

```
simplesh> cwd ; cd directorio ; cwd ; cd .. ; cwd
simplesh: cwd: /home/usuario
simplesh: cwd: /home/usuario/directorio
simplesh: cwd: /home/usuario
```

El comando interno cd

- (0.5p) **Opcional**: Implementa `cd [-]`:

```
simplesh> cd -
run_cd: Variable $OLDPWD no está definida
simplesh> cwd
simplesh: cwd: /home/usuario/directorio
simplesh> cd
simplesh> cwd
simplesh: cwd: /home/usuario
simplesh> cd -
simplesh> cwd
simplesh: cwd: /home/usuario/directorio
```

- El directorio por defecto es el valor de la variable de entorno `$HOME`, mientras que `$OLDPWD` contiene el directorio de trabajo previo
- Llamadas `POSIX` y/o `glibc` a tener en cuenta: `getenv()`, `setenv()` y `unsetenv()`
- Verifica que `simplesh` modifica el directorio actual siempre que encuentra el comando interno `cd`

```
simplesh> cwd ; cd ; cwd ; cd - ; cwd
simplesh: cwd: /home/usuario/directorio
simplesh: cwd: /home/usuario
simplesh: cwd: /home/usuario/directorio
```