

SISTEMAS OPERATIVOS II
Tercer curso Ingeniería Técnica de Sistemas.
Curso 2005-2006

Práctica 4: Procesos en UNIX: prioridades. Sistema de ficheros en UNIX. Redirección.

Añadir al intérprete de comandos de las prácticas anteriores las siguientes funciones

- **getpri** [**pid**] Devuelve la prioridad del proceso de identificador *pid*. Si no se especifica *pid* devuelve la prioridad del intérprete de comandos.
- **setpri** [**pid**] [**valor**] Establece la prioridad del proceso de identificador *pid* igual a *valor*. Si no se especifica *pid* establece la prioridad del intérprete de comandos. Si no se especifica ni *pid* ni *valor* muestra la prioridad del intérprete de comandos.
- **getlim** [**pid**] Devuelve el límite de la prioridad del proceso de identificador *pid*. Si no se especifica *pid* devuelve el límite de la prioridad del intérprete de comandos.
- **setlim** [**pid**] [**valor**] Establece el límite de la prioridad del proceso de identificador *pid* igual a *valor*. Si no se especifica *pid* establece el límite de la prioridad del intérprete de comandos. Si no se especifica ni *pid* ni *valor* muestra el límite de la prioridad del intérprete de comandos.
- **priexec** **valor** [**LISTA**] **com** [**&**] Ejecuta el comando descrito por *com* (programa y argumentos) estableciendo previamente su prioridad a *valor*. *LISTA* es opcional y representa una lista de variables de entorno.
 - Dependiendo de si las máquinas disponibles para realizar las prácticas tienen instalado Linux o Solaris, las funciones de obtener y establecer prioridades se implementarán con *getpriority* y *setpriority* o con *priocntl* respectivamente, lo que condicionará también el rango de valores de la prioridad. Las funciones *getlim* y *setlim* solo si implementarán si las máquinas disponibles para las prácticas tienen instalado Solaris.
 - Las funciones de ejecutar con prioridad cambiada deben ser compatibles con la ejecución en primer y segundo plano, tanto con lista de variables de entorno como sin ellas.
 - ejemplo:

```
#priexec -12 TERM DISPLAY HOME xterm -e /bin/csh &
```
- [**LISTA**] **com** <**fich** [**&**] Ejecuta el comando *com* (ejecutable con parámetros) con la entrada estándar redireccionada al fichero *fich*.

- **[LISTA] com >fich [&]** Ejecuta el comando *com* (ejecutable con parámetros) con la salida estándar redireccionada al fichero *fich*
 - **[LISTA]com %fich [&]** Ejecuta el comando *com* (ejecutable con parámetros) con el error estándar redireccionada al fichero *fich*.
 - **pipe com1 * com2** Ejecuta *com1* (ejecutable con parámetros) redireccionando su salida estándar a la entrada estándar de *com2* (ejecutable con parámetros).
 - Los comandos de redirección *>*, *<*, *%* crean un proceso para ejecutar *com* y deben ser compatibles entre sí con la ejecución en segundo plano, con la ejecución con las prioridades cambiadas, y con la especificación de un entorno.
- #priexec -15 TERM HOME a.out parm1 parm2 <f1 >f2 %f3 &
 ejecuta en segundo plano *a.out parm1 parm2 parm3* con la entrada, la salida y el error estándar redireccionado, además establece su prioridad a -15.

Información detallada de las llamadas al sistema y las funciones de la librería debe obtenerse con man (getpriority, setpriority, nice, priocntl..)

FORMA DE ENTREGA

Como en las prácticas anteriores

FECHA DE ENTREGA VIERNES 3 FEBRERO DE 2006