SOLUCIÓN EXAMEN MAYO 2013

1. Explique **razonadamente** si las siguientes afirmaciones son verdaderas o falsas:

I) (*1 p*) En la interfaz nodo-v/sfv de un sistema UNIX un nodo-v asociado a un fichero contiene entre otros datos un puntero al objeto descriptor de fichero.

II) (*1 p*) El algoritmo `issig()` del núcleo de un sistema UNIX se encarga, entre otras funciones, de invocar, si es necesario, al algoritmo del núcleo `sendsig()`.

Solución:

I) Un nodo-v, entre otras informaciones, contiene únicamente punteros a objetos dependientes del sistema de ficheros al que pertenece el archivo al que está asociado el nodo-v. Puesto que el objeto descriptor de fichero es una estructura independiente del sistema de ficheros la afirmación es **FALSA**.

II) En UNIX, el algoritmo del núcleo `sendsig()` es invocado por el algoritmo del núcleo `psig()` no por el algoritmo `issig()`. En conclusión la afirmación es **FALSA**.

2. Conteste **razonadamente** a las siguientes cuestiones relativas a un sistema UNIX SVR3:
a) (*1 p*) ¿Qué es el área de usuario o área U?
b) (*1 p*) Enumere la información contenida en el área U.

Solución:

a) El área de usuario o área U es una estructura local asociada a cada proceso que contiene información de control relevante sobre el mismo que el núcleo necesita consultar únicamente cuando ejecuta dicho proceso.

b) Entre la información contenida en los campos del área U se encuentra la siguiente:

- Un puntero a la entrada de la tabla de procesos asociada a dicho proceso.
- Los identificadores de usuario (uid, euid) y de grupo (gid, egid) del proceso.
- Los argumentos de entrada, los valores de retorno y el identificador del error (si se produjese) de la llamada al sistema en ejecución.
- Las direcciones de los manipuladores de las señales y otras informaciones relacionadas.
- Información acerca de las regiones de código, datos y pila obtenida de la cabecera del programa.
- La tabla de descriptores de ficheros que mantiene información sobre los ficheros abiertos por el proceso.
- El directorio de trabajo actual y el directorio raíz actual.
- El terminal de acceso asociada con el proceso, si existe alguno.
- Estadísticas de uso de la CPU.

3. Conteste **razonadamente** a las siguientes cuestiones relativas a un sistema UNIX SVR3:
a) (*1 p*) ¿Cuál es la función del ladrón de páginas?
b) (*1 p*) Enumere los cinco estados en que puede encontrarse una página asociada a una referencia de memoria legal que provoca un fallo de validez.

Solución:

a) El *ladrón de páginas* es un proceso del núcleo que se encarga de transferir al dispositivo de intercambio las páginas que ya no forman parte del conjunto de trabajo de un proceso. El núcleo crea al ladrón de páginas durante la inicialización del sistema y lo invoca cuando disminuye el número de páginas físicas libres.

b) La página que provocó el fallo se encontrará en uno de los siguientes cinco estados:

- Fuera de memoria principal alojada en un dispositivo de intercambio.
- Fuera de memoria principal en un fichero ejecutable en el disco.
- En la lista de marcos de páginas libres de memoria principal.
- Marcada como DZ (*Demand Zero*).
- Marcada como DF (*Demand Fill*).

4. Explique **razonadamente** el significado de las siguientes órdenes de un intérprete de comandos de UNIX:
a) (*0.5 p*) `echo $suma`
b) (*0.5 p*) `bg`
c) (*0.5 p*) `sort < procA >> procB`

Solución:

a) Muestra en el dispositivo de salida estándar, por defecto la pantalla, el valor de la variable suma

b) Lanza en segundo plano una tarea que hubiera sido previamente parada o suspendida.

c) Escribe al final del archivo `procB` el contenido del archivo `procA` ordenado alfabéticamente.

5. La máscara de modo simbólica del archivo ejecutable `j2013` que resulta de compilar el programa que se muestra en la Figura 1 es `-rwsrwxrwx` y dicho archivo pertenece al usuario `Paco` (UID=720, GID=103). El archivo `fichero1.txt` pertenece también al usuario `Paco` y su máscara de simbólica es `-rw-------`. El archivo `fichero2.txt` pertenece al usuario `David` (UID=725, GID=105) y su máscara de simbólica es `-rw-------`.

a) *(1 p)* Explique el significado de las sentencias enumeradas (**[ ]**) del código del ejecutable `j2013`.

b) Explique **razonadamente** el funcionamiento de este programa e indique los mensajes que aparecen en la salida estándar en los siguientes casos:

1) (*0.75 p*) El programa lo ejecuta `Paco`.

2) (*0.75 p*) El programa lo ejecuta el usuario `David`.

```
main()
{
    int x,y,fd1,fd2,fd3,fd4;

[1] x=getuid();
[2] y=geteuid();

    printf("\n[1] S1= %d, S2= %d",x,y);

[3] fd1=open("fichero1.txt", O_RDONLY);
    fd2=open("fichero2.txt", O_RDONLY);
    printf("\n[2] fd1= %d, fd2= %d", fd1, fd2);

[4] setuid(x);
    printf("\n[3] S1= %d, S2= %d",getuid(),geteuid());

    fd3=open("fichero1.txt", O_RDONLY);
    fd4=open("fichero2.txt", O_RDONLY);
    printf("\n[4] fd3= %d, fd4= %d", fd3, fd4);

    setuid(y);
    printf("\n[5] S1= %d, S2= %d\n",getuid(),geteuid());
}
```

Figura 1 – Código C del ejecutable `j2013`

Solución:

a) El significado de cada una de las sentencias marcadas con **[]** de este código es el siguiente:

[1] Llamada al sistema `getuid` para obtener el identificador de usuario real (*uid*) del proceso. Si se ejecuta con éxito en la variable `x` se almacena el *uid*. En caso contrario se almacena el valor -1.

[2] Llamada al sistema `geteuid` para obtener el identificador de usuario efectivo (*euid*) del proceso. Si se ejecuta con éxito en la variable `y` se almacena el *euid*. En caso contrario se almacena el valor -1.

[3] Llamada al sistema `open` para abrir el archivo `fichero1.txt` en modo de solo lectura. Si se ejecuta con éxito en la variable `fd` se almacena el descriptor del archivo. En caso contrario se almacena el valor -1.

[4] Llamada al sistema `setuid` para asignar el valor `x` al *euid* y al *uid* del proceso que invoca la llamada. Si el *euid* del proceso que invoca la llamada corresponde al del superusuario, entonces *euid*=`x` y *uid*=`x`. En caso contrario entonces *euid*=`x` si se cumplen algunas de las siguientes dos condiciones:

- `x` coindice con el valor del *uid* del proceso.
- La llamada al sistema `setuid` se está invocando dentro de la ejecución de un programa que tiene su bit `S_ISUID` activado y `x` coindice con el *uid* del propietario del programa.

b1) El usuario Paco (*uid*=720) es el propietario del archivo `j2013`. De acuerdo con su máscara de modo simbólica el propietario del archivo puede ejecutar el archivo. Al proceso A que se crea en el sistema asociado a la ejecución de `j2013` se le asigna *uid*=720 y *euid*=720. Al ejecutar el proceso A en primer lugar se invoca a la llamada al sistema `getuid` para asignar el valor del *uid* del proceso a la variable `x`. Luego se invoca a la llamada al sistema `geteuid` para asignar el valor del *euid* del proceso a la variable `y`. A continuación se imprime en la salida estándar, por defecto la pantalla, el mensaje

```
[1] S1= 720, S2= 720
```

Posteriormente se invoca a la llamada al sistema `open` para abrir con permiso de solo lectura el archivo `fichero1.txt`. La máscara de modo de este archivo únicamente permite leerlo al propietario del mismo, que es el usuario `Paco` cuyo *uid*=720. Puesto que el proceso A tiene un *euid*=720 que coincide con el *uid* del propietario del archivo la llamada al sistema se ejecuta con éxito y en `fd1` se almacena el descriptor del archivo.

Luego se vuelve a invocar a la llamada al sistema `open` para abrir con permiso de solo lectura el archivo `fichero2.txt`. La máscara de modo de este archivo únicamente permite leerlo al propietario del mismo, que es el usuario `David` cuyo *uid*=725. Puesto que el proceso A tiene un *euid*=720 que es distinto del *uid* del propietario del archivo la llamada al sistema devuelve un error y en `fd2` se almacena el valor -1. A continuación se imprime en la salida estándar el mensaje

```
[2] fd1= 3, fd2= -1
```

Posteriormente se invoca a la llamada al sistema `setuid` para asignar al *uid* y al *euid* del proceso A el valor de `x` (720). Como el *euid* del proceso A no es el del superusuario y el valor de `x` coincide con el valor del *uid* del proceso A entonces solamente el *euid* del proceso A se configura con el valor de `x`.

En resumen el resultado de esta llamada al sistema es configurar el *euid* del proceso A al valor *euid*=720, valor que por otra parte ya tenía sin necesidad de haber ejecutado esta llamada al sistema. A continuación se imprime en la salida estándar el mensaje

```
[3] S1= 720, S2= 720
```

Luego se invoca a la llamada al sistema `open` para abrir con permiso de solo lectura el archivo `fichero1.txt`. La máscara de modo de este archivo únicamente permite leerlo al propietario del mismo, que es el usuario Paco cuyo *uid*=720. Puesto que el proceso A tiene un *euid*=720 que coincide con el *uid* del propietario del archivo la llamada al sistema se ejecuta con éxito y en `fd3` se almacena el descriptor del archivo.

Luego se vuelve a invocar a la llamada al sistema open para abrir con permiso de solo lectura el archivo `fichero2.txt`. La máscara de modo de este archivo únicamente permite leerlo al propietario del mismo, que es el usuario `David` cuyo *uid*=725. Puesto que el proceso A tiene un *euid*=720 que es distinto del *uid* del propietario del archivo la llamada al sistema devuelve un error y en `fd4` se almacena el valor -1. Acto seguido se imprime en la salida estándar, el mensaje

```
[4] fd3= 4, fd4= -1
```

Posteriormente se invoca de nuevo a la llamada al sistema `setuid` para asignar al *uid* y al *euid* del proceso A el valor de `y` (720). Como el *euid* del proceso A no es el del superusuario y el valor de y coincide con el valor del *uid* del proceso A entonces solamente el *euid* del proceso

A se configura con el valor de `y`. En resumen el resultado de esta llamada al sistema es configurar el *euid* del proceso A al valor *euid*=720, valor que por otra parte ya tenía sin necesidad de haber ejecutado esta llamada al sistema. Finalmente se imprime en la salida estándar, el mensaje

```
[5] S1= 720, S2= 720
```

B2) Como la máscara de modo simbólica del archivo `j2013` tiene el bit `S_ISUID` activado entonces al ejecutarlo el usuario `David` al proceso A que se crea en el sistema asociado a la ejecución de `j2013` se le asigna un *euid* igual a *uid* del propietario del archivo, que recuérdese que es `Paco` *uid*=720. Luego el proceso A tiene *uid*=725 y *euid*=720.

Al ejecutar el proceso A en primer lugar se invoca a la llamada al sistema `getuid` para asignar el valor del *uid* del proceso a la variable x. Luego se invoca a la llamada al sistema `geteuid` para asignar el valor del *euid* del proceso a la variable y. A continuación se imprime en la salida estándar, por defecto la pantalla, el mensaje

```
[1] S1= 725, S2= 720
```

Posteriormente se invoca a la llamada al sistema `open` para abrir con permiso de solo lectura el archivo `fichero1.txt`. La máscara de modo de este archivo únicamente permite leerlo al propietario del mismo, que es el usuario Paco cuyo *uid*=720. Puesto que el proceso A tiene un *euid*=720 que coincide con el *uid* del propietario del archivo la llamada al sistema se ejecuta con éxito y en `fd1` se almacena el descriptor del archivo.

Luego se vuelve a invocar a la llamada al sistema `open` para abrir con permiso de solo lectura el archivo `fichero2.txt`. La máscara de modo de este archivo únicamente permite leerlo al propietario del mismo, que es el usuario `David` cuyo *uid*=725. Puesto que el proceso A tiene un *euid*=720 que es distinto del *uid* del propietario del archivo la llamada al sistema devuelve un error y en `fd2` se almacena el valor -1. A continuación se imprime en la salida estándar el mensaje

```
[2] fd1= 3, fd2= -1
```

Posteriormente se invoca a la llamada al sistema `setuid` para asignar al *uid* y al *euid* del proceso A el valor de `x` (725). Como el *euid* del proceso A no es el del superusuario y el valor de `x` coincide con el valor del *uid* del proceso A entonces solamente el *euid* del proceso A se configura con el valor de `x`. En resumen el resultado de esta llamada al sistema es configurar

el *euid* del proceso A al valor *euid*=725. A continuación se imprime en la salida estándar el mensaje

```
[3] S1= 725, S2= 725
```

Luego se invoca a la llamada al sistema `open` para abrir con permiso de solo lectura el archivo `fichero1.txt`. La máscara de modo de este archivo únicamente permite leerlo al propietario del mismo, que es el usuario `Paco` cuyo *uid*=720. Puesto que el proceso A tiene un *euid*=725 que es distinto del *uid* del propietario del archivo la llamada al sistema devuelve un error y en `fd3` se almacena el valor -1.

A continuación se invoca a la llamada al sistema `open` para abrir con permiso de solo lectura el archivo `fichero2.txt`. La máscara de modo de este archivo únicamente permite leerlo al propietario del mismo, que es el usuario `David` cuyo *uid*=725. Puesto que el proceso A tiene un *euid*=725 que coincide con el *uid* del propietario del archivo la llamada al sistema se ejecuta con éxito y en `fd4` se almacena el descriptor del archivo. Acto seguido se imprime en la salida estándar, el mensaje

```
[4] fd3= -1, fd4= 4
```

Posteriormente se invoca de nuevo a la llamada al sistema `setuid` para asignar al *uid* y al *euid* del proceso A el valor de `y` (720). Como el valor de `y` coincide con el valor del *uid* del propietario de `j2013` y el bits `S_ISUID` está activado entonces el *euid* del proceso A se configura con el valor de y, *euid*=720. Finalmente se imprime en la salida estándar, el mensaje

```
[5] S1= 725, S2= 720
```