Universidad Tecnológica Nacional
Facultad Regional Buenos Aires
Departamento de Ingeniería Electrónica

Técnicas Digitales III
6 de Diciembre de 2011
Final

Apellido y Nombres	Legajo	Nº Hojas	Profesor

1) Device Drivers

a) Mencione los tipos de "Device Drivers" de Linux, detallando sus diferencias.

b) En el caso de los controladores modulares, ¿qué instrucciones conoce para su incorporación al núcleo del S.O., su verificación y su remoción desde consola?

c) Describa con el mayor detalle posible la estructura de un "char device driver". Estructuras, declaraciones, macros, variables y funciones, que mínimamente debe contener.

d) Explique el mecanismo de bloqueo de procesos, que debe utilizar un controlador de dispositivos tipo carácter, empleado para esperar la recepción de un carácter.

e)

i. Explique la interfaz utilizada en núcleo de Linux versión 2.6, por el programador de aplicaciones, para acceder a las funciones implementadas en un controlador de dispositivos de caracteres.

ii. Detalle todas las estructuras utilizadas por el mecanismo indicado en el punto anterior.

f) La siguiente declaración corresponde al prototipo de la función invocada por el método *.read* de un controlador.

ssize_t *[Func Read]* (struct file *flip, char *buff, size_t count, loff_t *offp)

Indique:

ı. A qué espacio de memoria corresponde *buff.*

ıı. A qué espacio de memoria corresponde *flip.*

ııı. Si se produce un cambio de contexto o de nivel privilegio. Justifique.

---

2) IPC's

a) ¿Qué es un "named-pipe"?, ¿Cómo funciona?, ¿Cómo se accede?

b) Enumere los IPCs implementados por "System V". Explique sus características y usos normales. Qué caractateriza a las IPC's implementados por "System V".

c) Un proceso recibe de forma asincrónica y aleatoria datos desde un "socket" TCP (ya conectado) y por un "pipe" (ya creado). Escriba el código

Apellido y Nombres	Legajo	Nº Hojas	Profesor

necesario para recibir la información. Los datos no se reciben en un orden preestablecido.

d) Si en el punto (c) se reemplazara el "pipe" por una cola de mensajes. ¿Es posible utilizar la misma solución? Justifique.

e) ¿Cómo implementa a una aplicación que recibe de forma asincrónica y aleatoria datos desde un "socket" TCP y por una cola de mensajes?. Escriba el código necesario para recibir la información. Los datos no se reciben en un orden preestablecido.

---

3) Ejecución fuera de orden. Dibuje el esquema con las partes constitutivas y explique como se consigue ejecutar en diferente orden al flujo de instrucciones y como se reordenan los resultados. Traducción a micro operaciones. Composición de la Unidad de Ejecución(diferentes tipos de puertos de ejecución). ¿Cuál es la diferencia entre un procesador con un pipeline de ejecución, y uno que implemente un motor de ejecución fuera de orden?

---

4) Redes

a) ¿Qué función cumple la función listen()?, ¿En qué casos se utiliza?

b) ¿Qué función cumple la función bind()?, ¿En qué casos se utiliza?

c) ¿Siempre que se utiliza bind() también se utiliza listen()?

d) ¿Qué función debe utilizar para recibir conexiones UDP a través puerto determinado?

---

5) Escriba el assembler que binarice una imagen en escala de grises de ocho bits. El procedimiento recibe:

a) Considerando que el código en assember será invocado desde una rutina en C definida como:

```
void Binariza(unsigned char umbral, int tam, int puntero);
```

Nota: Binarizar una imagen es reemplazarla por una del mismo tamaño pero cuyos pixeles toman dos valores posibles: negro absoluto o blanco absoluto (0 o 255 respectivamente), de acuerdo al valor del pixel original respecto de uno definido como umbral. Los valores por debajo del umbral saturan a 0 y los que lo superan saturan a 255.

Considere el tamaño de umbral de un byte; tam y puntero de cuatro bytes.