

Informe del Proyecto Nº1

Sistemas Operativos
Universidad Nacional del Sur
2018

Integrantes:
Landau, Nicolás.
Puhl, Juan Manuel.

EXPERIMENTACIÓN DE PROCESOS Y THREADS CON LOS SISTEMAS OPERATIVOS.

Observación general: Todos los ejercicios de programación se realizaron y testearon en Linux Fedora 28.

Ejercicio 1.

1. Sudoku.

Se hicieron dos versiones, una utilizando procesos y otra con threads. Dado un archivo llamado sudoku.txt, se verificará si la jugada es correcta o no y se imprimirá por pantalla el resultado de la verificación.

A) Se crearon 27 procesos vinculados por medio de pipes, de esta forma podrían sincronizarse fácilmente. Luego se dividió la verificación en distintos métodos, en los cuales se verificaba un apartado distinto del sudoku en cuestión, por ejemplo, las subsecciones 3x3, las columnas y las filas.

B) Para la versión con threads, se utilizaron 27 hilos para cada subsección 3x3, columna y fila, de forma que cada uno compute de forma separada la correctitud del sudoku y maximizar la concurrencia. .

2. MiniShell.

Los métodos implementados son los siguientes:

- creaF: Crea un archivo en la dirección pasada por argumento. Al final de la dirección debe incluirse el nombre del archivo.
- creaD: Crea un directorio en la dirección pasada por argumento. Al final de la dirección debe incluirse el nombre del directorio.
- showF: Muestra por pantalla el contenido del archivo en la dirección pasada por argumento. Al final de la dirección debe incluirse el nombre del archivo.
- showD: Muestra por pantalla el contenido del directorio en la dirección pasada por argumento. Al final de la dirección debe incluirse el nombre del directorio.
- showM: Muestra por pantalla el contenido del manual de ayuda.
- elimF: Elimina el archivo en la dirección pasada por argumento. Al final de la dirección debe incluirse el nombre del archivo.
- elimD: Elimina el directorio en la dirección pasada por argumento. Al final de la dirección debe incluirse el nombre del directorio.
- exit: Finaliza la ejecución del shell.

Se eligieron estos nombres para cada uno de los métodos de forma que sean intuitivos y fácil de recordar y escribir, además se intento que todos tengan la misma cantidad de caracteres, para facilitar la programación.

Observación: En todos los casos menos en showM y exit, al ingresar el comando sin argumentos se utiliza una dirección predeterminada, la cual es /home/Nombre_usuario. Si se utiliza algún método de archivos, el nombre por defecto será Nuevo Archivo, si es un método de directorios, el nombre por defecto será Nueva Carpeta.

```

creaF: Crea un archivo de texto en el sistema, el usuario puede elegir la direcc
ion ingresandola como argumento.
elimF: Elimina un archivo de texto en el sistema.
showD: Muestra por pantalla el contenido de un directorio dado.
showF: Muestra por pantalla el contenido de un archivo dado.
showM: Muestra informacion sobre los posibles comandos.
exit: Finaliza la ejecucion del shell.

-----

>>creaF home/Juanma/Documentos/Archivo_de_prueba.txt
Archivo creado con exito.
>>elimF home/Juanma/Documentos/Archivo_de_prueba.txt
Eliminado correctamente.
>>elimF home/Juanma/Documentos/Archivo_de_prueba.txt
No se ha podido eliminar el archivo.
>>creaD home/Juanma/Directorio_de_prueba
Directorio creado con exito.
>>elimD home/Juanma/Directorio
No se ha podido eliminar el directorio.
>>elimD home/Juanma/Directorio_de_prueba
Directorio eliminado con exito.
>>showF home/Juanma/Documentos/Archivo
No se ha podido abrir el archivo.
>>showF home/Juanma/Documentos/archivo.txt
-----Contenido-----
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla at imperdiet ex.

```

Ejercicio 2

1. Sincronización de impresoras.

Para realizar este inciso se utilizaron semáforos e hilos. Ambas impresoras se implementaron como semáforos mientras que los usuarios se representaron como hilos. Se imprime por pantalla el estado de cada usuario o hilo.

```

estoy imprimiendo en la impresora 1.
estoy imprimiendo en la impresora 2.
termine de imprimir en la impresora 1.
estoy imprimiendo en la impresora 1.
termine de imprimir en la impresora 2.
estoy imprimiendo en la impresora 2.
termine de imprimir en la impresora 1.
estoy imprimiendo en la impresora 1.
termine de imprimir en la impresora 2.
estoy imprimiendo en la impresora 2.
termine de imprimir en la impresora 1.
estoy imprimiendo en la impresora 1.
termine de imprimir en la impresora 2.
estoy imprimiendo en la impresora 2.
termine de imprimir en la impresora 1.
estoy imprimiendo en la impresora 2.
termine de imprimir en la impresora 2.
estoy imprimiendo en la impresora 1.
estoy imprimiendo en la impresora 2.
termine de imprimir en la impresora 1.
estoy imprimiendo en la impresora 2.
estoy imprimiendo en la impresora 1.
termine de imprimir en la impresora 2.
termine de imprimir en la impresora 1.

```

2. Sincronización de impresoras con prioridad.

Para la resolución de este inciso se utilizaron semáforos contadores y binarios e hilos. Ambas impresoras se implementaron como semáforos binarios y cada usuario como un hilo. Además se utilizó un hilo extra a modo de controlador de prioridades que se ejecuta en todo momento y que vigila las colas de prioridad. Para llevar a cabo el manejo de prioridades se tuvieron en cuenta tres posibles valores, 0,1 y 2, siendo 2 la prioridad más alta. Los hilos se van almacenando en la cola correspondiente a su prioridad, y luego el controlador los manejaba, moviéndolos a una cola auxiliar de hilos listos para imprimir. Esto se hizo porque se quiso que la impresión ocurriera dentro del método del hilo y no fuera.

```
[1] VENGO A IMPRIMIR. Mi prioridad es: 1.
[0] VENGO A IMPRIMIR. Mi prioridad es: 1.
[8] VENGO A IMPRIMIR. Mi prioridad es: 0.
[9] VENGO A IMPRIMIR. Mi prioridad es: 1.
[4] VENGO A IMPRIMIR. Mi prioridad es: 2.
[5] VENGO A IMPRIMIR. Mi prioridad es: 1.
[6] VENGO A IMPRIMIR. Mi prioridad es: 1.
[12] VENGO A IMPRIMIR. Mi prioridad es: 2.
[13] VENGO A IMPRIMIR. Mi prioridad es: 1.
[14] VENGO A IMPRIMIR. Mi prioridad es: 2.
[15] VENGO A IMPRIMIR. Mi prioridad es: 1.
[16] VENGO A IMPRIMIR. Mi prioridad es: 0.
[11] VENGO A IMPRIMIR. Mi prioridad es: 1.
[10] VENGO A IMPRIMIR. Mi prioridad es: 2.
[7] VENGO A IMPRIMIR. Mi prioridad es: 0.
[3] VENGO A IMPRIMIR. Mi prioridad es: 1.
[2] VENGO A IMPRIMIR. Mi prioridad es: 0.
[19] VENGO A IMPRIMIR. Mi prioridad es: 1.
[18] VENGO A IMPRIMIR. Mi prioridad es: 1.
[17] VENGO A IMPRIMIR. Mi prioridad es: 0.
La impresora 1 esta libre.
[4] Hay impresoras libres. Voy a imprimir.
[4] Estoy imprimiendo en la impresora nro 1.
La impresora 2 esta libre.
[12] Hay impresoras libres. Voy a imprimir.
[12] Estoy imprimiendo en la impresora nro 2.
[4] Ya termine de imprimir en la impresora nro 1.
La impresora 1 esta libre.
[14] Hay impresoras libres. Voy a imprimir.
[14] Estoy imprimiendo en la impresora nro 1.
[12] Ya termine de imprimir en la impresora nro 2.
La impresora 2 esta libre.
[10] Hay impresoras libres. Voy a imprimir.
[10] Estoy imprimiendo en la impresora nro 2.
[14] Ya termine de imprimir en la impresora nro 1.
La impresora 1 esta libre.
[1] Hay impresoras libres. Voy a imprimir.
[1] Estoy imprimiendo en la impresora nro 1.
```

Ejercicio 2:

Asistente: Para este ejercicio se utilizaron hilos para representar al asistente y a cada estudiante, además de semáforos para cada una de las 3 sillas. Por pantalla se muestra el estado actual del asistente y se avisa si algún estudiante debió irse debido a que no había lugar.

```
[nicolas@localhost Proyecto]$ ./alumnosAyudante.sh
Consultando...
Consulta terminada
Consultando...
Consulta terminada
Consultando...
Consulta terminada
Consultando...
Un estudiante se fue porque estaban las 3 sillas ocupadas.
Consulta terminada
Consultando...
Un estudiante se fue porque estaban las 3 sillas ocupadas.
Consulta terminada
Consultando...
Consulta terminada
Consultando...
Consulta terminada
Consultando...
Consulta terminada
```

PROBLEMAS

Ejercicio 2:

a) En este artículo se habla sobre cómo Intel lleva a cabo las migraciones de sistemas operativos, teniendo en cuenta el cambio de modelo, en el cual se opta por ofrecer sistemas operativos como servicio y ofrecer gran cantidad de actualizaciones pequeñas, frente a la anterior, de una única actualización grande. Es por esto que Intel decidió modificar su modo de actuar e ideó una serie de buenas prácticas a tener en cuenta.

Evaluación del SO:

Buena Práctica #1: Testear el SO.

Testear el SO en una muestra representativa de los dispositivos disponibles para conocer cómo funcionaría en cada caso. Con este propósito, el proceso de testeo se centra en la estabilidad, desempeño y experiencia del usuario, debido a que estos tres aspectos tienen el mayor impacto en la productividad y satisfacción de los empleados. También se evalúan otros aspectos, tales como la integración del SO con la estructura de soporte de IT, redes y conectividad, almacenamiento en la nube y sincronización, etc.

Buena Práctica #2: Establecer el valor para el negocio del SO.

Durante esta etapa, se intenta encontrar ventajas a largo plazo, para ello se evalúa el SO en términos de productividad (viendo cómo puede mejorar el rendimiento de usuarios), costo general, recuperación de inversión (es decir, evaluar costos y beneficios del SO en la empresa) y seguridad (ver qué características pueden ofrecer más seguridad a los usuarios dentro de la empresa).

Buena Práctica #3: Considerar la demanda del usuario.

Es importante saber qué opinan los empleados como usuarios del sistema. Los empleados se sentirán más seguros si utilizan un SO que conocen y que les agrada.

Preparación para la migración:

Buena Práctica #4: Plan para la preparación operacional. (Entrega y soporte).

Verificar que se puede entregar el SO tanto en los casos de actualización, como de instalación en nuevas pc's. También se evalúan los cambios a llevar a cabo para soportar el nuevo SO durante y después de la migración.

Buena Practica #5: Verificar preparación de las aplicaciones.

Es necesario verificar que cada una de estas soporta el nuevo SO y mitigar el impacto de aquellas que no son totalmente compatibles.

Para esto pueden usarse maquinas virtuales que tengan el SO nuevo instalado, y ejecutar las aplicaciones allí y testearlas, para luego notificar a los respectivos dueños y pedirles que actualicen. Es recomendable hacer esto con al menos las aplicaciones mas utilizadas dentro de la empresa, sin embargo es bueno testear la mayor cantidad de aplicaciones posibles.

Buena Practica #6: Completar tareas ingenieriles.

Crear y testear los medios para desplegar el SO, es decir, la imagen del sistema, la cual contiene el SO, drivers y aplicaciones principales. Es importante probar que funcione cada uno de sus componentes, la seguridad, y el manejo de componentes y políticas de la imagen. También puede hacerse seguimiento y tener en cuenta las opiniones de usuarios para mejorarla.

Buena Practica #7: Iniciar entrenamiento.

Es necesario disponer de gente capacitada en el nuevo SO.

1. Usuarios Finales: Ofrecer una comunidad de soporte online para los primeros usuarios del SO. Esto puede hacerse usando FAQ's, videos, preguntas interactivas, una plataforma de red social donde los empleados puedan discutir problemas y compartir información, etc.
2. Personal de soporte: Implementar un curso online de entrenamiento en el nuevo SO. También es bueno crear artículos para ayudar al personal. Darles acceso temprano por medio de MV's y/o laptops.

Buena Practica #8: Comunicarse regularmente.

Es importante comunicar los cambios en la empresa:

- Comunicación con empleados: Nombrar a una persona como la encargada de las comunicaciones, lo cual ayuda a crear un flujo controlado de comunicaciones.
- Comunicación con clientes internos e inversores: Crear un equipo de representantes de clientes, incluyendo un sub-equipo dedicado a manejar las relaciones con cada grupo empresarial. En cuanto a los inversores, es bueno mantenerlos al día con reportes semanales y presentaciones cada dos semanas.

Buena Practica #9: Integrar herramientas de inteligencia empresarial.

Usar herramientas de terceros para proveer información relevante que pueda ayudar a rastrear progreso y tomar decisiones durante el proceso de entrega del SO.

Buena Practica #10: Dividir en capas el entorno de computo.

Usar técnicas de división en bloques, lo cual permite deconstruir el entorno de computo en varias capas de aplicaciones, contenido y personalización. De esta forma se pueden hacer cambios en una sola capa sin alterar a las demás.

Despliegue.

Buena Practica #11: Desplegar en fases.

1. Fase 1: Permitir que ciertos usuarios utilicen el sistema en una fase beta, de forma que puedan ayudar al proceso de testeo. Generalmente esta etapa involucra a unos 500 usuarios.
2. Fase 2: Aumentar el despliegue a mas gente, en especifico, desarrolladores y testers involucrados en el proceso de preparación de aplicaciones. Involucra a unos 2000 usuarios.
3. Fase 3: Ofrecer el nuevo SO como opción estándar. en las pc's. De esta forma, los empleados o empleadores pueden elegir al recibir una nueva computadora si quieren el nuevo SO o el anterior.
4. Fase 4: Es la etapa final del despliegue, en la cual el nuevo SO se convierte en el estándar.

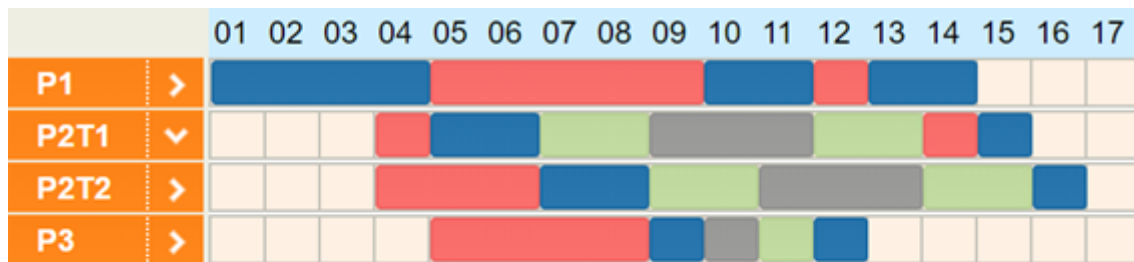
Buena Practica #12: Usar una amplia gama de métodos de migración.

Es decir, ofrecer varias opciones de actualización, así como instaladores de un solo clic, o gente que se dedique exclusivamente a esto. Un ejemplo es Microsoft. A la hora de lanzar el Windows 10 ofreció en su pagina oficial un método de actualización gratuito.

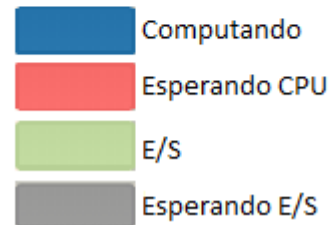
b) Con respecto a nuestra carrera, Ingeniería en Computación, creemos que este articulo es muy interesante ya que en un futuro, es probable que nuestro trabajo se trate de esto, y ademas nos permite conocer todas las fases de trabajo y que hay que adaptarse a los cambios en el ámbito de la computación. En el futuro es posible que hayan otros modelos nuevos de SO, y seamos nosotros los encargados de establecer estas buenas practicas y llevarlas a cabo.

Ejercicio 3

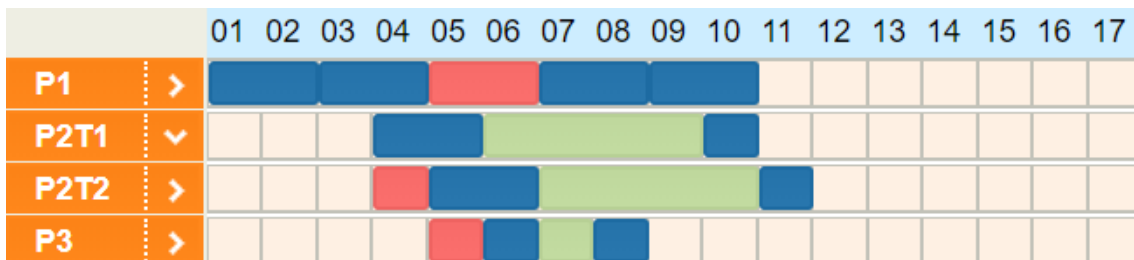
a) Round-Robin con un procesador.



Proceso	Tiempo de Retorno	Tiempo de Espera
P1	14	6
P2T1	12	5
P2T2	13	6
P3	8	5

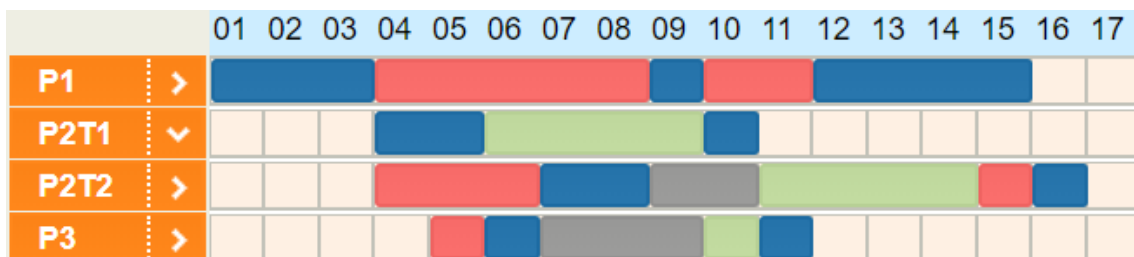


b) Round-Robin con dos procesadores.



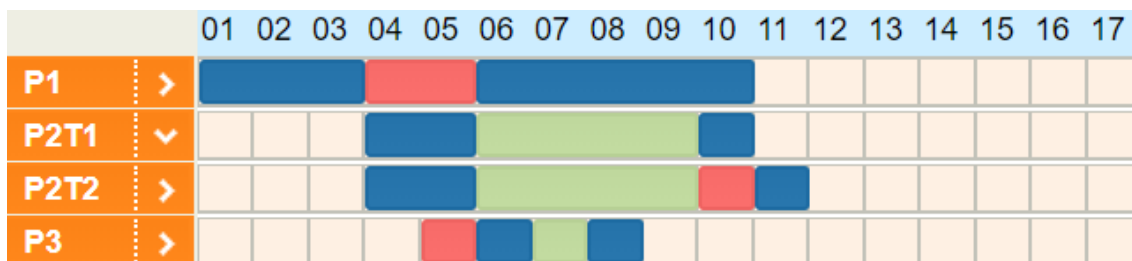
Proceso	Tiempo de retorno	Tiempo de espera
P1	10	2
P2T1	7	0
P2T2	8	1
P3	4	1

c) SJF un procesador, un dispositivo de E/S



Proceso	Tiempo de retorno	Tiempo de espera
P1	15	7
P2T1	7	0
P2T2	13	6
P3	7	4

d) SJF dos procesadores, múltiples dispositivos de E/S



Proceso	Tiempo de retorno	Tiempo de espera
P1	10	2
P2T1	7	0
P2T2	8	1
P3	4	1