TEC | Tecnológico de Costa Rica

Instituto Tecnológico de Costa Rica
Área Académica de Ingeniería en Computadores
CE 4303 — Principios de Sistemas Operativos

Proyecto 1
Comunicación de procesos sincronizada

Fecha de asignación:	5,6 de Setiembre, 2023	Fecha de entrega:	26,27 de Setiembre, 2023
Grupos:	4-3 personas	Profesor:	Jason Leitón, Luis Barboza

1. Objetivo

Aplicar técnicas comunicación de procesos, con el fin de transferir información entre procesos pesados, utilizando memoria compartida.

2. Atributos a evaluar

- Aprendizaje continuo. Se requiere que el estudiante valore las estrategias y el conocimiento adquirido para alcanzar el objetivo.

3. Motivación

El problema de concurrencia y comunicación es un clásico de los conflictos en los ámbitos de los sistemas operativos, muchos de los problemas que se encuentran en la informática y otras áreas se puede modelar de la misma manera, y con ello se podría solucionar aplicando las mismas técnicas y principios. La idea fundamental de este proyecto es comunicar *heavy process* que se ejecutan en núcleos separados a través de memoria principal . Esta sección de memoria deberá ser inicializada de alguna manera.

4. Descripción

La idea general de este proyecto es comunicar y sincronizar distintos procesos pesados a través de memoria compartida. Estos procesos se estarán ejecutando en núcleos separados.

En el proyecto existen 2 tipos de procesos, los cuales deben ser ejecutados en CPU distintos, el primero se debe de ejecutar en el Cortex-A9 de la FPGA y el otro debe ejecutarse en el que diseñará cada grupo utilizan Qsys. A continuación se detalla cada uno de ellos:

4.1. Proceso en el Cortex-A9

Se debe de quemar alguna imagen de algún sistema operativo basado en linux para que sea ejecutado en la fpga. Para este apartado se preparó un tutorial que se puede ven el siguiente

link: https://drive.google.com/file/d/1k5BE5TuLOIbNAT090eEDCVBiyqVmtCuF/view. Una vez que el sistema operativo se haya iniciado, se deberá ejecutar el proceso correspondiente a través de serial (pueden utilizar SSH). Este proceso será el encargado de tomar un archivo de texto ubicado en el sistema de archivos del SO y lo leerá para pasar los datos al núcleo de Qsys. El contenido del archivo corresponde a un conjunto de valores de píxeles, separados por comas, el cual es guardado por filas. Estos valores serán cifrados utilizando el algoritmo RSA con una clave de 16 bits (esta clave se le debe hacer padding en caso de que no se encuentre de manera completa).

Este proceso debe tomar todos los pixeles y pasarlos (utilizando memoria) al CPU de Qsys, con el fin de que realice un descifrado de la información. Dependiendo del modo en el que se ejecute, este proceso también debe de procesar, sin embargo, más adelante se detallan los modos.

4.2. Proceso en Qsys

Se deberá de crear un CPU utilizando Platform Designer, en el cual se debe ejecutar un proceso que se encargará de descifrar la información que proviene del proceso anterior. Es importante mencionar que las claves deben ser de 16 bits, las cuales deben ser proporcionados por los botones y mostrados (en hexa) en todos los 7segmentos, una vez que las claves son colocadas (una tras otra) se presionará otro botón para dar inicio a la lectura de los píxeles y con ello descifrarlos. Cada valor descifrado debe mostrarse en los 7 segmentos en decimal. Al final del descifrado se debe de guardar (en la sd donde está el SO) un archivo tipo imagen, para posteriormente visualizarlo.

Una vez que la información haya sido decifrada, se debe de aplicar el filtro laplaciano, este proceso se debe hacer apenas se tenga los datos para iniciar con el procesamiento . Posteriormente se debe guardar la imagen original y la filtrada para ser mostradas.

4.3. Modos de ejecución

Se tendrán dos modos de ejecución los cuales se indicarán por medio de un switch, estos modos son:

1. Manual: En ese este modo se necesita que el usuario presione un botón para descifrar un píxel que ya está en memoria. Cada vez que se presiona el botón se debe descifrar el siguiente píxel y mostrarse en el 7segmentos. En caso de que se deje presionado el botón, el proceso descifrará píxeles sin parar. Cabe destacar que en cualquier instante el usuario puede cambiar a modo automática para que continue el procesamiento sin tener que presionar un botón.

2. Automático. El procesamiento de los datos se realiza de manera continua sin interacción del usuario. Sin embargo, en cualquier instante el usuario puede cambiar al modo manual y no debe generar ningún problema.

4.4. Procesamiento heterogéneo

Esta sección solo aplica para la parte de aplicar el filtro. El sistema debe ser capaz de elegir cuanto porcentaje (el mecanismo de elección del porcentaje es libre para cada grupo) de los píxeles se realizaran en el CPU de Qsys y cuantos se realizaran en el Cortex. Por ejemplo, si el usuario indica que 50 % en Qsys, entonces la mitad de la imagen se procesa en Qsys, mientrás que la otra en el Cortex, posteriormente se debe unir el resultado. Se debe tener en cuenta que no tiene que aparecer marcas en la imagen, tales como lineas horizontales o un trozo de imagen sin completar. Al finalizar el proceso se debe mostrar en los 7 segmentos en segundos cuanto fue la duración total de la aplicación descifrado + filtro.

4.5. Consideraciones de implementación

1. El algoritmo de cifrado a utilizar es RSA, lo que significa que se debe de descifrar utilizando la siguiente fórmula: $NewP = OldP^d \%n$ donde d y n son la llave privada.

2. Las claves serán de 16 bits, se deben introducir utilizando los botones y los 7 segmentos (se aconseja que sea en hexadecimal).

3. Los datos cifrados serán separados por comas y se guardaran por filas de píxeles.

4. En tecdigital se muestra un ejemplo de archivo cifrado. Además, del resultado tanto del descifrado, como la del filtro.

5. Los dos primeros números elementos corresponde a las dimensiones de la imagen (ancho x alto).

6. Se recomienda utilizar mmap y una memoria para datos en Qsys.

5. Puntos extras

Los grupos que integren mostrar las imágenes reconstruidas y filtradas por medio de VGA (Se puede utilizar cualquier componente del catálogo) se acreditarán 10 puntos extras.

6. Requerimientos técnicos

- Este proyecto se debe realizar en el lenguaje de programación C o Rust.
- Debe ser implementado en Linux (no máquina virtual) y se debe proporcionar un makefile.
- No se permite soluciones “alambradas”.

- Se debe prestar especial atención a los errores de acceso a memoria o utilización de recursos. Es **inaceptable** el error *segmentation fault* o *core dumped*.
- No se permite **busy waiting** como mecanismo para evitar condiciones de carrera.

7. Documentación- Estilo IEEE-Trans (máximo 5 páginas)

- Introducción: Teoría necesaria, breve descripción del proyecto y qué es lo que se espera en el escrito.
- Ambiente de desarrollo: Configuración básica se debe utilizar para ejecutar el proyecto. Frameworks, bibliotecas externas o principales, aplicaciones de terceros, herramientas de desarrollo.
- Atributos: Esta sección deben de describirse cuales atributos fueron reforzados durante el desarrollo del proyecto. Para el atributo de **aprendizaje continuo** debe responder las siguientes preguntas:
  - ¿Cuales son las necesidades actuales de aprendizaje para enfrentar el proyecto?
  - ¿Cuáles son las tecnologías que se pueden utilizar para el desarrollo?
  - ¿Cuáles acciones se implementó para el desarrollo del proyecto (organización de tiempo, búsqueda de información, repaso de contenidos, entre otros)?
  - Evalúe de forma crítica la eficiencia de las acciones implementadas en el contexto tecnológico.

  Para el atributo de **Herramientas de ingeniería** se requiere indicar lo siguiente:
  - ¿Cuales herramientas, bibliotecas o recursos se utilizaron en el proyecto?
  - ¿Cómo se aplicaron los recursos o bibliotecas seleccionados?
  - Indique cómo adaptó los recursos en general para desarrollar el proyecto
- Detalles del diseño del programa desarrollo, tanto del software como del hardware (en caso de que aplique): Diagramas de flujo, imágenes, descripciones entre otros, todo lo que sea necesario para entender de una mejor manera el diseño y funcionamiento del proyecto. Es necesario que realice como mínimo, el diagrama de arquitectura, componentes y secuencia.
- Instrucciones de cómo se utiliza el proyecto.
- Tabla de actividades por cada estudiante: bitácora con el total de horas trabajadas por estudiante.

- Conclusiones
- Sugerencias y recomendaciones.
- Referencias

8. Entregables

- Código fuente con documentación interna.
- Documentación.
- Archivos necesarios para ejecutar el programa.

9. Evaluación

- Algoritmo de filtro 10 %
- Transferencia de datos 10 %
- Integración 10 %
- Sincronización 10 %
- Algoritmo de descifrado 10 %
- Display y modo de ejecución 10 %
- Documentación 20 %

10. Fecha de entrega

- 26, 27 de Setiembre 23:55 por tecdigital

11. Otros aspectos administrativos

- Cualquier duda sobre la especificación debe de ser aclarada con el profesor o asistente.
- Para la revisión del proyecto se debe de entregar tanto la documentación como la implementación del software.
- No se reciben trabajos después de la hora indicada.

- En la revisión del proyecto pueden estar presentes el coordinador y asistente.
- Es responsabilidad del estudiante proveer los medios para poder revisar la funcionalidad del software, por ejemplo, si no se realiza la interfaz, se debe de proporcionar otro medio para la verificación, de lo contrario la nota será cero en los rubros correspondientes a la funcionalidad faltante.
- Cualquier ambigüedad o duda puede contactar al profesor para aclararla.
- En caso de que necesite algún requerimiento de hardware o software coméntelo con el profesor con anticipación.