Máster universitario en sistemas espaciales. Gestión de datos

Examen de junio (31 de mayo de 2021)

Apellidos, Nombre MATAIX CABALLERO, DIEGO Nº de Matrícula

Responda en esta misma hoja, utilizando únicamente el espacio asignado para cada pregunta.

1 Describa requisitos específicos de los computadores (hardware y software) a bordo de un satélite.

- Funcionamiento en tiempo real: ha de ser capaz de procesar datos/instrucciones y ejecutar programas en tiempo real. La asignación de recursos es esencial para estos sistemas.
- Resistencia al ambiente: ha de ser "radiation-hardened" para soportar las condiciones de operación en el espacio. También tiene que resistir las vibraciones y cargas a las que está sometido
- Computador embebido: se emplean computadores embebidos con OS de tiempo real ya que se requiere una capacidad
- Alta fiabilidad: al ser sistemas altamente costosos y la imposibilidad de reparar el sistema, se requiere que tengan alta fiabilidad y que los sistemas hayan sido verificados y validados para uso espacial.

Estos requisitos suelen resultar en el uso de computadores bastante menos potentes que los que se usan para otras labores, como sería un ordenador de sobremesa.

2 Considere una pantalla de 480x272 pixels que representa cada pixel con un formato RGB, en donde se representan los colores mediante una combinación de intensidades de los colores primarios Rojo (R), Verde (G) y Azul (B). Cada una de las intensidades de los colores primarios se representa con 8 bits.

1. ¿Cuántos colores diferentes se pueden representar con ese modelo RGB?

R - 8 bits
G - 8 bits
B - 8 bits

combinaciones entre los colores disponibles

, cada combinación de 8 bits por color da 2^8 intensidades posibles/color

, por lo tanto para RGB, las combinaciones son $2^8 \cdot 2^8 \cdot 2^8 = 16777216$ posibles colores $= 2^{24}$

2. ¿Cuántos bytes se necesitan para representar todos los pixels de la pantalla?

~~8 bits colores 8 bits/pixel~~ ; 480 x 272 pixels = 130 560 pixels

=> 8 bits · 8 bits · 8 bits = 512 bits por pixel.

=> 512 bits/pixel · 130 560 pixels = 66846720 bits = 8.36 MB

3 ¿Para qué sirve una tabla de vectores de interrupciones? La respuesta debe incluir:

1. de qué tipo son los elementos de la tabla.
2. de qué orden es el tamaño de esa tabla, y de que depende ese tamaño.
3. quién inicializa el contenido de la tabla y con qué valores.

Las interrupciones son instrucciones que envía el sistema para interrumpir un proceso en ejecución.
Es esencial para las labores de 'scheduling' y asignar recursos de manera efectiva.
Una tabla de vectores de interrupciones contiene los 'interrupts' que tiene que enviar el sistema.

4 Las unidades de compilación (como paquetes o funciones) pueden tener dos tipos: especificación y cuerpo. Describa estos elementos en Ada.

- especificación (.ads): donde se declaran las variables que se van a emplear.
  (.h en C) De esta manera se consigue un nivel de abstracción mayor que facilita el trabajo con un gran número de ficheros.

- cuerpo (.adb): donde se programan las tareas y funciones.
  (.c en C) Hace una llamada a la especificación (.ads), normalmente al principio del código.

5 Los sistemas operativos de propósito general proporcionan un conjunto de servicios básicos. Sin embargo, en los sistemas operativos empotrados a bordo de un satélite no es aconsejable incluir todos estos servicios. Describa los servicios básicos que es imprescindibles que estén incluidos en un sistema operativo en un satélite.

Los servicios básicos de un OS para un satélite son:

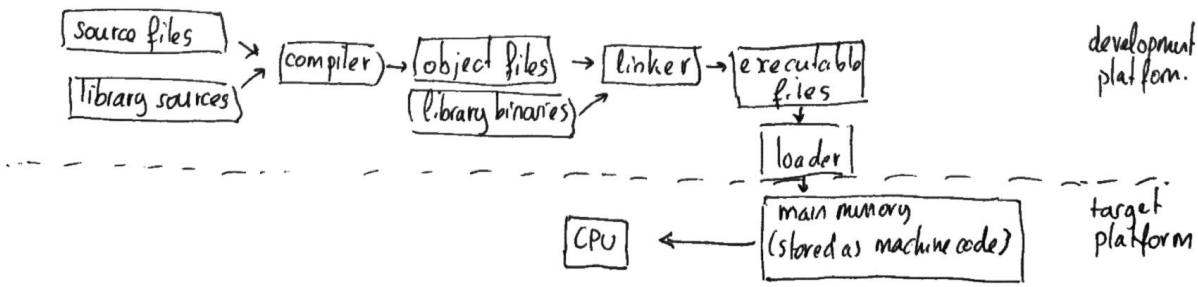
1) Gestión de ficheros: Funcionalidad básica para gestionar los ficheros en los que guarda datos el satélite.
2) Gestión de recursos: Funcionalidades básicas de asignación de memoria, sistema de scheduling y tasking.
3) ejecución de programas: Únicamente al iniciarse, donde ejecuta un programa (que se ejecuta continuamente)
4) comunicaciones: Funcionalidades básicas
5) Módulos I/O: Necesario en un satélite para comunicarse con los periféricos (sensores, actuadores...)
6) Detección de errores: Necesario para detectar errores del sistema, o en la ejecución del programa.
7) Interfaz de usuario (No es imprescindible en OS satélite)
8) Protección y seguridad (No es imprescindible en OS satélite)
9) Contabilidad: Gestión de la info. sobre recursos empleados y sin utilizar.

6 Describa brevemente los componentes de un entorno para desarrollo de software cruzado y su diferencia con un nativo.

~~El entorno de compilación desarrollo del software no es el mismo que el entorno en el que se desea ejecutar el software (SW), entonces entonces para cargar que~~

Si el entorno de desarrollo de SW no es el mismo que el entorno en el que se desea ejecutar el SW se requieren herramientas de compilación cruzada. Esto sucede ya que en el desarrollo de SW embebido, la plataforma donde se ejecuta el SW no suele contar con teclado o alguna forma de visualizar la ejecución del programa. Por lo tanto, en la compilación cruzada se desea compilar el código en un lenguaje máquina distinto al que comprende el procesador de la plataforma donde se está desarrollando, por lo que se suele hacer uso de librerías específicas (por ejemplo para Spark).

La compilación cruzada sigue el siguiente proceso:



7 Describa cuál es el objetivo de un planificador de tiempo real y algún método de planificación.

El objetivo de un planificador en tiempo real es el de asignar las tareas de manera que se puedan ejecutar de manera secuencial. De esta forma le llega al procesador una programa en ejecución por proceso, o varias tareas si se emplean varias 'threads' (hebras).
Hay dos tipos principales:

- Round-Robin: asigna la ejecución de las tareas según la disponibilidad de recursos. Por ejemplo en un PC.
- Priority based: asigna la ejecución de las tareas según la prioridad asignada a estas. Este modelo es esencial para sistemas de tiempo real críticos, por ejemplo en un satélite.
  Cuando una tarea con mayor prioridad que la que se está ejecutando es requerida, el sistema manda un interrupt y la tarea pasa a ejecutarse (mientras la que estaba ejecutando deja de hacerlo y se encuentra aún en modo 'active' y 'runnable'.

8 En la práctica de laboratorio de software de a bordo se ha usado un Conversor Analógico Digital (ADC). Describa con qué objetivo y qué componente de la arquitectura de software lo usa.

Un ADC se emplea para convertir señales analógicas, usualmente voltaje, a señales digitales que puedan ser interpretadas por el procesador. Su uso es esencial a la hora de emplear un gran número de sensores. Por ejemplo, al conectar una IMU (con señal analógica) a una Raspberry Pi es necesario un ADC, pero no lo es para una Arduino que tiene puertos analógicos.

En la práctica de laboratorio este componente se encontraba en la misma placa que utilizábamos, ya que esta cuenta con una serie de sensores.

Los componentes de los módulos I/O pueden emplear ADCs (como en la Arduino).

9 Enumere tres tipos de requisitos no funcionales, y cómo se aplica cada uno de ellos en un ejemplo de requisito no funcional.

Los requisitos no funcionales son aquellos que no se refieren a la funcionalidad del sistema si no que se refieren a sus propiedades. Ejemplos de este tipo de requisitos según las reglas ECSS-Q-ST-80 son la capacidad de cómputo del procesador, (FLOP y operaciones con complemento a) la capacidad de memoria, la seguridad y mantenibilidad o la 'reliability'.

10 En la verificación y validación se suelen usar herramientas de análisis dinámico del código. Describa brevemente en qué consiste este tipo de software y algunos tipos las métricas obtenidas. (NCET) ARS

Las herramientas de análisis dinámico del código examinan el código ejecutándolo y determinando si su funcionamiento es correcto, por lo que se prueban las distintas rutas del código (bloques if-else, loops, for...). Además, se analiza el tiempo de ejecución del código en distintos casos. Se emplean métricas como el WCET (worst-case execution time) que determina el peor tiempo de ejecución y modelos como el de Ravenscar. No se puede llevar acabo un análisis de verificación y validación completo, por lo que cada métrica tiene inconvenientes.

NOTAS: 8 de junio de 2021 DURACIÓN: 60 minutos
REVISIÓN: 10 de junio de 2021 PUNTUACIÓN: cada pregunta vale un punto.