



SISTEMAS OPERATIVOS II



TEMA 2

Conceptos de Sistemas Operativos: El caso Unix

Área de Arquitectura y Tecnología de Computadores

Escuela Universitaria Politécnica de Teruel

<http://?????.?????.es/SOII/>

1

Conceptos de S. O.: El caso Unix

Bibliografía:

- Silberschatz & Galvin: Sistemas Operativos
- Stallings: Sistemas Operativos
- Robbins: Unix programación práctica
 - Capítulos 1, 2 y 3
- Ayuda de Unix: “man”

2

Conceptos de S. O.: El caso Unix

Objetivos:

- Recordar y afianzar algunos conceptos clave de los S. O. relacionados con procesos, planificación de la CPU, ficheros y gestión de memoria

3

Conceptos de S. O.: El caso Unix

Parte 1: **Procesos**

Parte 2: **Planificación de la CPU**

Parte 3: **Ficheros**

Parte 4: **Gestión de memoria**

4

Conceptos de S. O.: El caso Unix

Parte 2: Planificación de la CPU

- Introducción
- Conceptos básicos
- Tipos de planificación
- Criterios del planificador a corto plazo
- Algoritmos de planificación
- Planificación en múltiples procesadores
- Planificación en Tiempo-Real
- Evaluación de algoritmos

5

Planificación de la CPU

Objetivos:

- Recordar los principales criterios de planificación de la CPU
- Repasar los principales algoritmos de planificación en un sistema monoprocesador
- Recordar la planificación en sistemas específicos:
 - Sistemas multi-procesadores
 - Sistemas de Tiempo Real

6

Introducción

- Multiprogramación:** consiste en tener un proceso en ejecución en todo momento
 - Finalidad: maximizar el aprovechamiento de la CPU
- Tiempo compartido:** consiste en compartir la CPU entre los procesos de usuario
 - Finalidad: permitir que todos los usuarios puedan interactuar con sus programas en ejecución

7

Introducción

¿Cuál es la base de la multiprogramación y del tiempo compartido?

- La planificación de la CPU
- El intercambio o la conmutación de la CPU entre procesos

Objetivo de la Planificación:

- Mínimo tiempo de respuesta
- Máximo throughput
- Máxima eficiencia del procesador

“Conseguir un computador más eficiente y productivo”

Conceptos básicos

- Def.- **recurso** es una entidad que se demanda o se consume.
- Los procesos necesitan recursos para evolucionar a lo largo del tiempo y realizar su tarea:
 - Ejemplos: tiempo de CPU
memoria
acceso a ficheros
dispositivos de E/S
- Los recursos se asignan a los procesos:
 - Cuando se crean
 - Durante la ejecución, según sus necesidades

9

Conceptos básicos

Tipos de recursos:

Recursos "preferentes"

- Una vez concedidos pueden ser utilizados por otros procesos aunque estos no hayan terminado de utilizarlos

Ejemplos: CPU, disco duro, etc.

Recursos "NO preferentes"

- Una vez concedidos no pueden ser utilizados por otros procesos hasta que el proceso al que se le asignaron haya finalizado

Ejemplos: espacio en ficheros, terminales, etc.

10

Conceptos básicos

Decisiones del S. O. relacionadas con recursos:

- ¿Qué proceso obtiene el recurso?
 - Asignación:** Dado un conjunto de peticiones de recursos, ¿a qué procesos hay que asignar los recursos solicitados con el objetivo de conseguir un uso más eficiente de dichos recursos?
- ¿Cuánto tiempo puede mantener un proceso los recursos asignados?
 - Planificación (Scheduling)
 - Planificación del procesador
 - Planificación de la memoria en sistemas de MV
 - Planificación de las operaciones de E/S

11

Conceptos básicos

Propiedad:

- La ejecución de un proceso se puede dividir en ráfagas de ejecución (CPU) y ráfagas de E/S (E/S)

La CPU es el recurso más importante en un sistema multiprogramado

- Deseamos maximizar la utilización de la CPU, ya que existen varios procesos en memoria a la vez

Solución:


- Cuando un proceso tiene que esperar una operación de E/S, la CPU se asigna a otro proceso que esté en estado preparado

12







Conceptos básicos


El “dispatcher”

 Es el módulo encargado de transferir el control de la CPU al proceso seleccionado por el planificador (scheduler)

Tareas del dispatcher:

-  Realizar los cambios de contexto
-  Realizar los cambios de modo (modo usuario / supervisor)
-  Actualizar PC con la posición adecuada del programa

 Debe estar programado de manera que sea lo más rápido posible


 Es invocado en cada cambio (conmutación) de contexto

13





Conceptos básicos

El “dispatcher”

 **Def.- latencia de “dispatch”**, tiempo transcurrido entre la parada de un proceso y el arranque de otro

Nota:


-  La existencia de interrupciones permite la multiprogramación
-  El tratamiento de interrupciones exige la existencia de rutinas especiales para atenderlas

14




Tipos de planificación


Planificación a largo plazo


 Decide sobre la creación de los nuevos procesos

Planificación a medio plazo

 Decide el “swapping”

Planificación a corto plazo

 Decide qué proceso será el siguiente al que se le asigne la CPU


 “Preparado” → “En Ejecución”

15



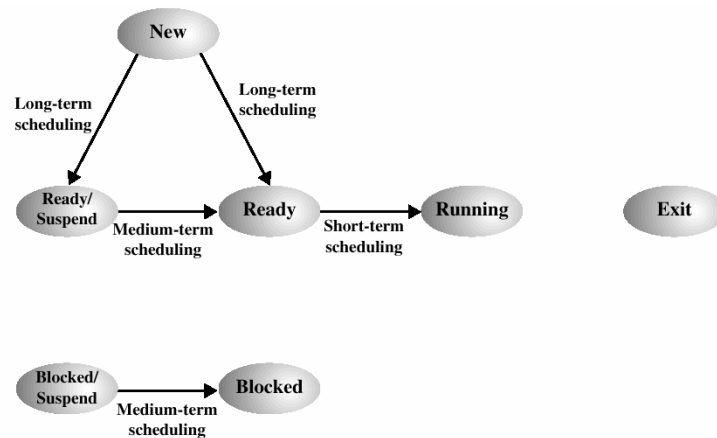
Tipos de planificación

Planificación de E/S

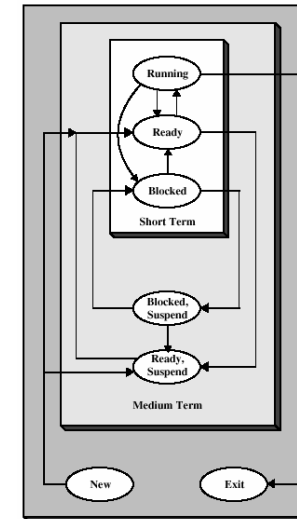
 Decide qué solicitud de E/S pendiente se tratará por un dispositivo disponible

16

Transiciones de estado de los procesos y planificación



Niveles de planificación



18

Tipos de planificación

Planificación a largo plazo

- ☞ Determina qué programas son admitidos en el sistema para ser procesados
- ☞ Controla el grado de multiprogramación
- ☞ A mayor cantidad de procesos, menor será el porcentaje de tiempo asignado a cada uno de ellos

19

Tipos de planificación

Planificación a medio plazo

- ☞ Forma parte de la función de "swapping"
- ☞ Permite manejar el grado de multiprogramación del sistema

20

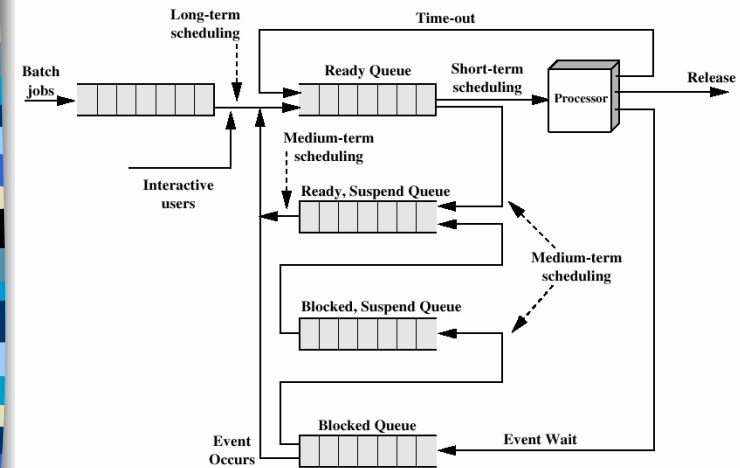
Tipos de planificación

Planificación a corto plazo

- Denominado "dispatcher"
- Se ejecuta frecuentemente
- Se invoca cuando ocurre un evento:
 - Interrupción de reloj
 - Interrupción de E/S
 - Llamada al sistema operativo (System calls)
 - Señales (Signals)

21

Diagrama de colas de planificación



Criterios del planificador a corto plazo

Orientados al usuario

- Tiempo de respuesta**
 - Tiempo desde que se emite una petición hasta que se recibe la primera respuesta del sistema
- Tiempo de retorno**
 - Tiempo desde que se emite una petición hasta que se completa su servicio
- Tiempo de espera**
 - Tiempo que el proceso espera en la cola de procesos listos
- Previsibilidad**
 - El tiempo de ejecución de un trabajo debe ser independiente de la carga del sistema

Criterios del planificador a corto plazo

Orientados al sistema

- Productividad: Throughput**
 - Maximizar el número de trabajos terminados por unidad de tiempo
- Utilización del procesador**
 - Maximizar el porcentaje de tiempo que el procesador está ocupado
- Recursos equilibrados**
 - Mantener ocupados todos los recursos del sistema
 - Favorecer a los procesos que menos recursos utilizan

24

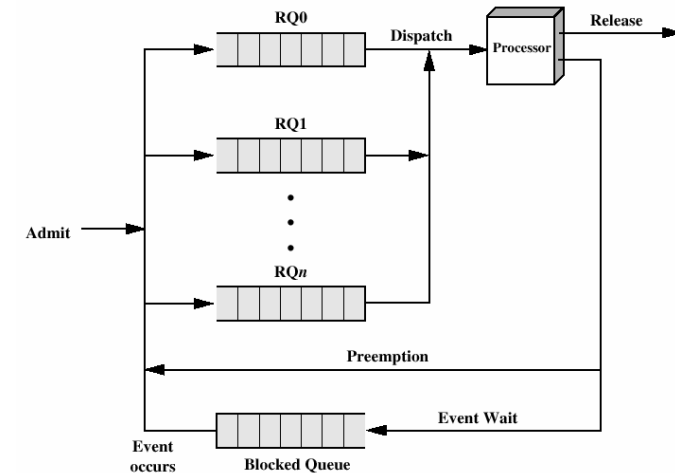
Criterios del planificador a corto plazo

Prioridades

- ☞ Cada proceso tiene asignada una prioridad
- ☞ El planificador elige siempre el proceso con mayor prioridad entre los preparados
- ☞ Existe una cola de preparados para cada prioridad
- ☞ Los procesos de baja prioridad pueden sufrir inanición
 - ☞ "Una solución": Permitir que los procesos cambien su prioridad en función de su historia en el sistema

25

Diagrama de colas de prioridad



Criterios del planificador a corto plazo

Otros

Plazos

- ☞ Cada tarea tiene asociado un plazo de finalización

Equidad

- ☞ En ausencia de otras directrices, los procesos deben tratarse de forma equitativa

27

Criterios del planificador a corto plazo

Modos de decisión

No preferente (No expropiativo)

- ☞ Cuando un proceso consigue la CPU, no la abandona hasta que termina o se bloquea en espera de una operación de E/S o un evento

Preferente (Expropiativo)

- ☞ Un proceso en estado de ejecución puede ser interrumpido por el S. O. pasando a estado "Preparado"
- ☞ Se evita que un proceso pueda monopolizar la utilización de la CPU durante mucho tiempo

28

Criterios del planificador a corto plazo

📁 Cambios de estado, modos de decisión y planificación

📁 ¿Cuándo debe decidir el planificador el proceso a ejecutar?

- 📁 Finalización de un proceso
- 📁 Cambio de estado: "En Ejecución" → "Bloqueado"
- 📁 Cambio de estado: "En Ejecución" → "Preparado"
- 📁 Modo de decisión preferente (expiración del quantum)
- 📁 Cambio de estado: "Bloqueado" → "Preparado"
- 📁 Modo de decisión preferente (prioridades)

29

Criterios del planificador a corto plazo

📁 Optimización de criterios de planificación:

- 📁 Maximizar la utilización de la CPU
- 📁 Maximizar el throughput
- 📁 Minimizar el tiempo de retorno
- 📁 Minimizar el tiempo de espera
- 📁 Minimizar el tiempo de respuesta

30

Algoritmos de planificación

- 📁 **FCFS**: First-Come, First-Served (\cong cola FIFO)
- 📁 **SJF**: Shortest Job First
- 📁 **SPN**: Shortest Process Next
- 📁 **SRT**: Shortest Remaining Time
- 📁 **HRRN**: Highest Response Ratio Next
- 📁 **RR**: Round-Robin
- 📁 **Colas con Realimentación**

31

Algoritmos de planificación







📁 **FCFS**: First-Come, First-Served (\cong cola FIFO)

- 📁 Cuando un proceso abandona la CPU, este algoritmo selecciona el proceso que lleva más tiempo en la cola de "Preparados". Dicho proceso será el que pase a ejecutarse
- 📁 Un proceso "corto" puede esperar durante mucho tiempo antes de ejecutarse
- 📁 Favorece los procesos limitados por CPU (CPU-bound)
 - 📁 Los procesos con ráfagas de E/S largas tendrán que esperar a que los procesos CPU-bound abandonen el procesador
 - 📁 Efecto convoy
- 📁 No preferente (No expropiativo)

32

Algoritmos de planificación




SJF: Shortest Job First **SPN**: Shortest Process Next

-  Este algoritmo selecciona el proceso con menor ráfaga de CPU. Dicho proceso será el que pase a ejecutarse
-  Los procesos cortos se adelantan a los largos
 -  El comportamiento de los procesos largos es menos previsible, pueden sufrir inanición
-  Debe estimarse el tiempo de la siguiente ráfaga de CPU del proceso
 -  Si el tiempo estimado para un proceso no es el correcto, el S. O. puede abortarlo
-  No preferente (No expropiativo)

33

Algoritmos de planificación



SRT: Shortest Remaining Time

-  Versión preferente de la política SPN
-  Se debe estimar el tiempo de la siguiente ráfaga de CPU del proceso
-  Preferente (Expropiativo)

34

Algoritmos de planificación








HRRN: Highest Response Ratio Next

-  Elige el siguiente proceso a ejecutar según el ratio:
$$\frac{\text{Tiempo de espera en las colas} + \text{Tiempo de servicio}}{\text{Tiempo de servicio}}$$
-  No preferente (No expropiativo)

35

Algoritmos de planificación

RR: Round-Robin

-  Utiliza una preferencia basada en el reloj
 -  Quantum: tiempo máximo de permanencia en ejecución que se asigna a cada proceso
-  Es el algoritmo utilizado en un S. O. de tiempo compartido
-  Depende mucho del tamaño del quantum
 -  Si el quantum es muy grande \cong FCFS
 -  Si el quantum es adecuado \cong estrategia compartir el procesador
 -  Si el quantum es muy pequeño, se debe considerar el efecto de los cambios de contexto

 Preferente

36

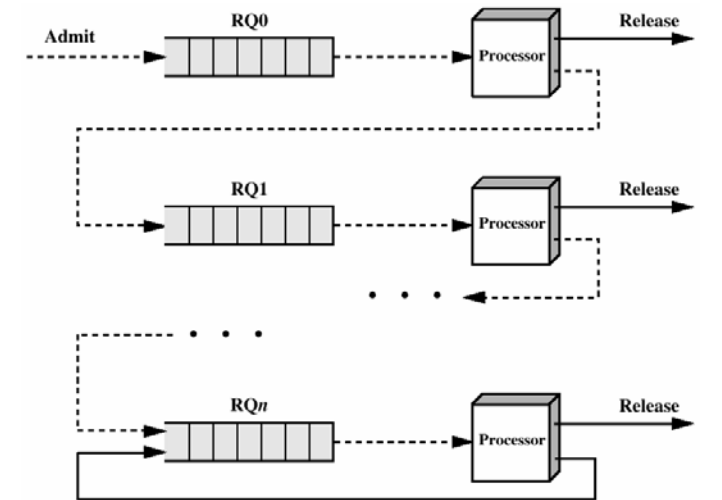
Algoritmos de planificación

Colas con Realimentación

- Varias colas con distinta prioridad
- Los procesos pasan de unas a otras en función de su comportamiento temporal
- Puede ser preferente (expropiativo) y no preferente (no expropiativo)

37

Colas con Realimentación



Otros algoritmos de planificación

Planificación por prioridades

- Se asocia una prioridad a cada proceso
- La CPU se asigna al proceso que tiene la prioridad más alta
- Definición de las prioridades:
 - **Internamente:** Se definen a partir de cantidades medibles
 - Límites de tiempo, requisitos de memoria, nº de ficheros abiertos, ratio entre la ráfaga media de E/S y la ráfaga media de CPU
 - **Externamente:** Se usan criterios externos al S. O.
 - Importancia del proceso, tipo y cantidad de los fondos para pagar el mantenimiento del computador, el departamento al que pertenece el propietario del proceso

Otros algoritmos de planificación

Planificación por prioridades

- Puede ser preferente y no preferente
- Problema: Inanición
 - Solución: Aumentar la prioridad de los procesos con el paso del tiempo
- **Nota:** El algoritmo SJF es un caso particular de éste algoritmo; la prioridad del proceso es el inverso de la duración de la siguiente ráfaga de CPU

40

Otros algoritmos de planificación

MQ: Multilevel Queue

- La cola de procesos “preparados” se divide en diferentes colas que permiten clasificar los procesos en grupos
- Cada proceso se asocia a una cola en función de sus características:
 - Prioridad, tamaño de memoria, tipo de proceso, etc.
- En cada una de las colas se puede aplicar un algoritmo de planificación diferente:
 - Ejemplo con dos colas:
 - Nivel 1: RR
 - Nivel 2: FCFS

41

Otros algoritmos de planificación

MQ: Multilevel Queue

Ejemplos:

1) Planificación por prioridades estáticas y preferente:

- 1º Procesos de sistema
- 2º Procesos interactivos
- 3º Procesos de edición interactiva
- 4º Procesos batch

2) Planificación por quantum de tiempo:

- Se asigna tiempo de CPU a cada una de las colas
- El tiempo asignado a cada cola se distribuye entre los procesos según el algoritmo de planificación de dicha cola

Ejemplo:

- 80% de CPU para cola de procesos interactivos con un RR
- 20% de CPU para cola de procesos batch con un FCFS

42

Otros algoritmos de planificación

MFQ: Multilevel-Feedback-Queue

- La cola de procesos “preparados” se divide en diferentes colas
- Los procesos se distribuyen por las colas según las características de sus ráfagas de CPU
- Un proceso puede cambiar de cola a lo largo de su ejecución:
 - Si un proceso utiliza mucho la CPU será trasladado a una cola de menor prioridad
 - Los procesos intensivos en E/S y los procesos interactivos se desplazarán hacia la cola de máxima prioridad
 - Un proceso que lleva en espera mucho tiempo será trasladado a una cola de mayor prioridad

Otros algoritmos de planificación

MFQ: Multilevel-Feedback-Queue

- Es el algoritmo de planificación más general, y por lo tanto, el más complejo
- Parámetros que definen el algoritmo MFQ:
 - Número de colas
 - Algoritmo de planificación para cada cola
 - Método utilizado para determinar cuándo promocionar un proceso
 - Método utilizado para determinar cuándo degradar un proceso
 - Método utilizado para determinar en qué cola entrará un proceso que necesita la CPU

44



Planificación en múltiples procesadores

- El problema de la planificación es más complejo y no existe una solución definitiva
- Problema:** Reparto de la carga de trabajo entre los diferentes procesadores
- Soluciones:**
 - Una única cola de procesos “preparados” para todos los procesadores
(Multiprocesamiento simétrico)
 - Problema:** Varios procesadores pueden intentar ejecutar el mismo proceso
 - Una cola de procesos “preparados” para cada uno de los procesadores
 - Problema:** Equilibrio de la carga entre los procesadores para maximizar su utilización



Planificación en múltiples procesadores

- Problema:** Reparto de la carga de trabajo entre los diferentes procesadores
- Mas soluciones:**
 - Multiprocesamiento **asimétrico**: estructura “Master-Slave”
 - Una única cola de procesos “preparados” para todos los procesadores
 - Un procesador hace de maestro (Master) y los demás hacen de esclavos (Slave)
 - El procesador maestro ejecuta código de S.O.
 - Los procesadores esclavos ejecutan código de usuario
 - Problema:** El procesador maestro se convierte en un cuello de botella

46



Planificación en Tiempo-Real

Tipos de Sistemas de Tiempo Real:

- Hard real-time system
 - Requieren la finalización de tareas críticas dentro de un intervalo estricto de tiempo
- Soft real-time system
 - Requieren que determinados procesos críticos sean prioritarios con respecto a otros procesos

47



Planificación en Tiempo-Real

Hard Real-Time Systems: Otras características

- Para cada proceso se debe especificar el tiempo de CPU que necesita para su finalización
- Nunca utilizan memoria virtual. En los sistemas con memoria virtual no se puede garantizar la finalización de un proceso en un tiempo especificado
- Los procesos críticos controlan hardware dedicado
- Los computadores y S. O.s modernos de propósito general no soportan tiempo real
- Ejemplo: **RTLinux, VxWorks**

48



Planificación en Tiempo-Real

📖 Soft Real-Time Systems: Otras características

📖 Problemas:

- 📖 La asignación no equitativa de recursos puede generar retrasos en las ejecuciones de algunos procesos

📖 Requisitos de implementación:

- 📖 El sistema debe tener un algoritmo de planificación por prioridades
- 📖 La prioridad de los procesos de tiempo real debe ser la más alta. Además estos procesos no deben degradarse
- 📖 La latencia del dispatch debe ser pequeña

49



Evaluación de algoritmos

📖 ¿Cómo seleccionar para un sistema concreto un algoritmo de planificación de la CPU?

📖 Problema 1: Definir criterios de selección de algoritmos

- 📖 Utilización de la CPU
- 📖 Tiempo de respuesta
- 📖 Throughput
- 📖 Es necesario definir la importancia relativa de cada medida

50



Evaluación de algoritmos

📖 ¿Cómo seleccionar para un sistema concreto un algoritmo de planificación de la CPU?

📖 Problema 2: Determinar el método de evaluación

- 📖 Modelado determinista
 - 📖 http://eupt2.unizar.es/ asignaturas/itq/sistemas_operativos_1/PFC.htm
- 📖 Modelos de colas
- 📖 Simulación
- 📖 Implementación

51