Sistemas Operativos II

LESI

Grupo de Sistemas Distribuídos

http://gsd.di.uminho.pt

As más notícias

- O exame 'quase' não tem componente teórica
- E a componente prática é só programação concorrente
- É uma cadeira de "engenharia": temos de
  - Perceber os compromissos => usar a "massa cinzenta"
  - Sujar as mãos na "massa" => programar, programar, programar

As boas notícias

- Todos estamos habituados a funcionar em ambientes de concorrência:
  - Fazemos várias coisas ao "mesmo" tempo
  - E sabemos que às vezes isso dá asneira

    => controlo de concorrência!
- A concorrência é 'apenas' mais uma forma de estruturação de programas. Percebido o esquema mental, é fácil...

Devo vir às aulas teóricas?

- Absolutamente!
- Temos 2 semanas de teoria com assuntos HOT (por exemplo RAID, sistemas de ficheiros robustos, distribuídos,...)
- E depois passamos o resto das aulas a resolver exercícios
  - INCLUINDO os dos exames anteriores

Equipa Docente

- Responsável pela disciplina + aulas teóricas
  - Francisco Soares de Moura (fsm@di.uminho.pt)
- Aulas práticas
  - Rui Oliveira, José Orlando Pereira, Vitor Fonte, Carlos Baquero, ...
- Horário de atendimento: terça 9:45-10:45 ?

Programa

- Sistemas de ficheiros
  - Requisitos, objectivos, estudo de casos
  - RAID, Log struct FS, sistemas de ficheiros distribuídos
- Recapitulação de conceitos de programação concorrente
  - Modelos de concorrência, comunicação e sincronização (processos e threads)
- Mãos na massa:
  - Exercícios de programação concorrente comprocessos, semáforos, threads, mutexes…
  - Aulas práticas em laboratório: Linux

Bibliografia recomendada

- Sebenta de Sistemas Operativos (em construção…)
- A. Silberschatz et al., *Applied Operating System Concepts*, John Wiley & Sons, 2000.

OU

- A. S. Tanenbaum, *Modern Operating Systems*, 2nd edition, Prentice Hall, 2001.

Bibliografia recomendada

- fsm 2004, ***Vou fazer Sistemas Operativos***
- www.google.com
  - Introduction to operating systems
  - …
- www.gildot.org, www.slashdot.org …

Bibliografia Adicional

- R. Stevens, *Advanced Programming in the Unix Environment*, Addison Wesley, 1990.
- Diversos artigos sobre sistemas operativos, a disponibilizar na página da cadeira ou a pesquisar na Internet.
- Manuais do sistema operativo, FAQs, código fonte do Linux, …

Transparências

- (Progressivamente) disponíveis em:

  http://gsd.di.uminho.pt/SOI/5305O3.html
- Baseadas nas transparências originais correspondentes aos livros recomendados
- Servem apenas como "âncora" ao estudo

Avaliação

- Exame final (+ pequena parte de avaliação nas práticas ?)
- Exame cobre sobretudo a matéria prática
  - Código, pequenos programas concorrentes
  - Valoriza-se a capacidade de raciocínio e a concepção de algoritmos (por oposição à utilização de "padrões" de soluções)
- Ninguém faz a disciplina apenas com a parte teórica

Gestão de Ficheiros

- Sistemas de ficheiros
  - Recapitulação de hw e sw de IO
    - Discos, partições, disk IO, device drivers, concorrência, caches, etc.
  - Requisitos, objectivos, estudo de casos
  - RAID, Log structured File Systems
  - Noções de sistemas de ficheiros distribuídos

Sistemas de ficheiros: requisitos

- Persistência
- Grande escala (quantidade de ficheiros + dimensão elevada)
- Rapidez de acesso (Tempo de acesso a disco >> TaccRAM)
- Concorrência
- ...

Objectivos (1)

- Armanzenamento
  - Persistente (backup, undelete, RAID)
  - Eficiente
    - Espaço (=> aproveitar)
      - Dados (exemplos)
        - Alocação não contígua para eliminar fragmentação externa
        - Suporte para ficheiros "dispersos" (resultado de "hash", por exemplo)
      - Metadados, eg. estruturas para representar blocos livres/ocupados: FAT, i-nodes, ...
    - Tempo: algoritmos de gestão e recuperação rápidos

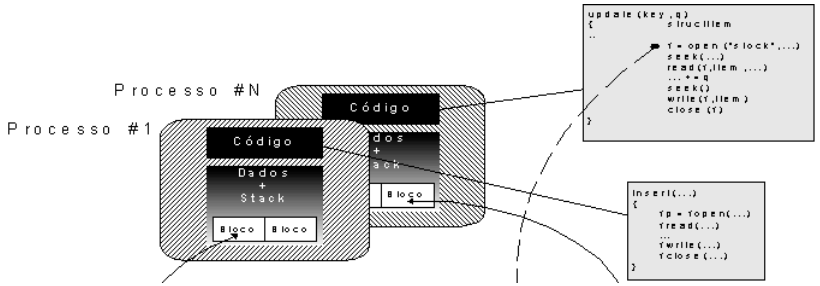
Objectivos (2)

- Acesso
  - Escalável
  - Conveniente
    - estrutura interna visivel (pelo kernel) ou só pelas aplicações?
      - Sequencia de bytes vs. Ficheiros indexados
  - Seguro
    - controlo de acessos
    - Auditoria
    - privacidade...

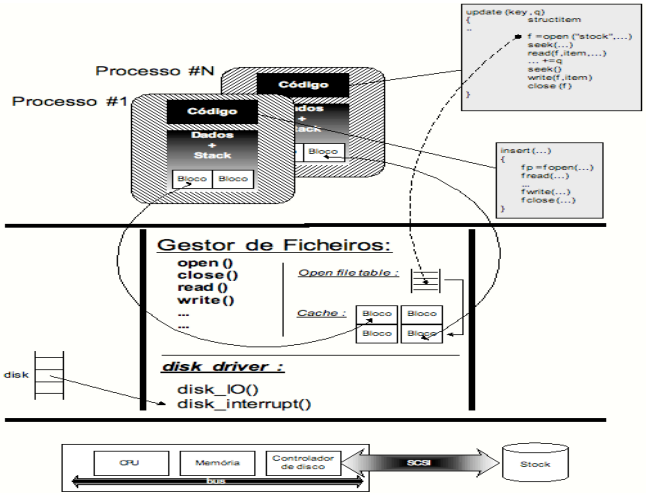
Objectivos (3)

- Acesso
  - Rápido (alguns exemplos de "bom-senso")
    - Evitar dispersão de blocos pelo disco => cuidado na alocação, usando por exemplo os "cilinder groups" do BSD, "file extents" do JFS e XFS
    - Escalonamento de pedidos de transferência para evitar movimentos do braço
    - Uso de caches (em disco e RAM) e delayed write => CUIDADO!
    - Directorias
      - Podem ter milhares de entradas (eg. e-mail!)
      - Procura sequencial? Binária? B-trees?

The big picture



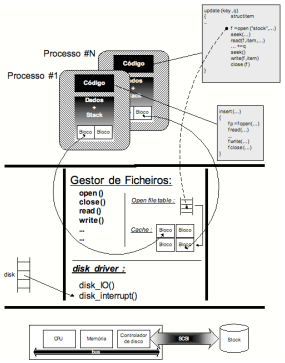
Dois programas a acederem simultaneamente ao mesmo ficheiro ou base de dados



The big picture revisited

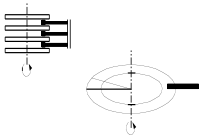
- Assegure-se que percebeu
  - Como surgem as "race conditions"
    - entre processos
    - dentro do SO
  - Vantagens/desvantagens do uso de caches

Note que estamos a falar de caches por software, cópias de dados em memória mas acessíveis em contextos diferentes



Discos

- O tempo necessário para aceder a um bloco é determinado por três factores:
  - Tempo de procura (posicionamento na pista)
  - Tempo de rotação do disco (posicionamento no sector)
  - Tempo de transferência
- O tempo de procura (seek) é dominante



Escalonamento de pedidos de transferência

- FIFO
- SSTF
- Elevator
- Scan circular

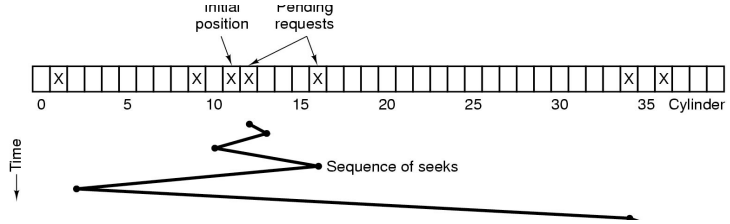
```
Disk_io()
{
Do_IO...
}
```

```
RTIntDisk()
{
...
Rti
}
```

Consegue imaginar os algoritmos?

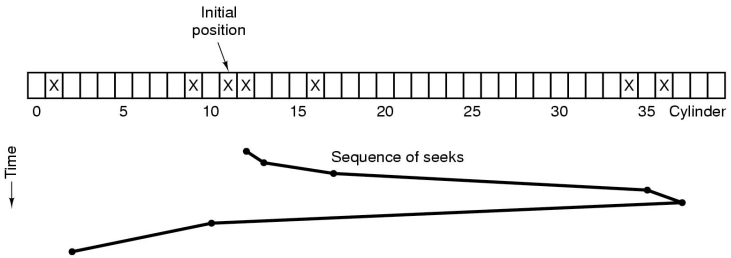
Como bloquear um processo até que chegue a vez do seu pedido?

Escalonamento de pedidos a disco



Shortest Seek First (SSF)

Escalonamento de pedidos a disco



Elevador

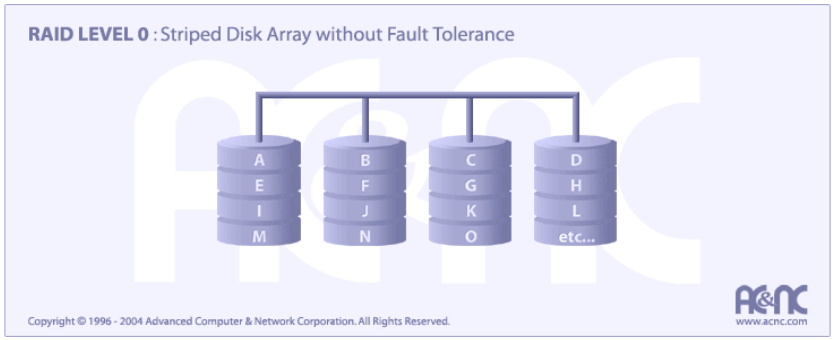
E se um disco tem uma avaria?

?

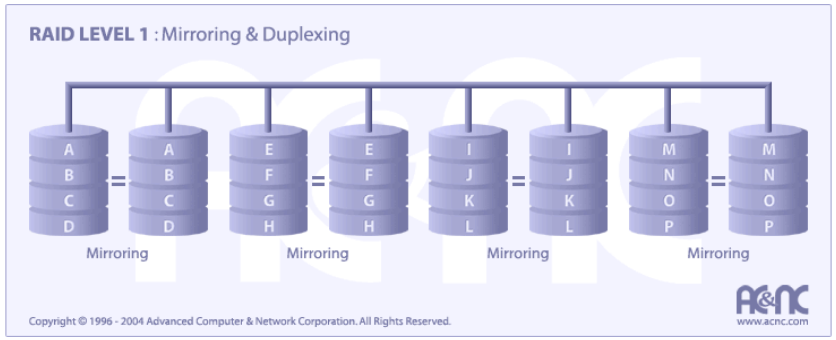
RAID

- Redundant Arrays of Inexpensive Disks
  - Pesquise no google por “Raid-1 Raid-5 primer”
- Objectivos:
  - Desempenho
  - Disponibilidade
    - Tolerância a faltas nos discos (depende do tipo de RAID)
    - Não resolve ficheiros apagados, virus, bugs, etc
      - Continua a precisar de BACKUPs!!

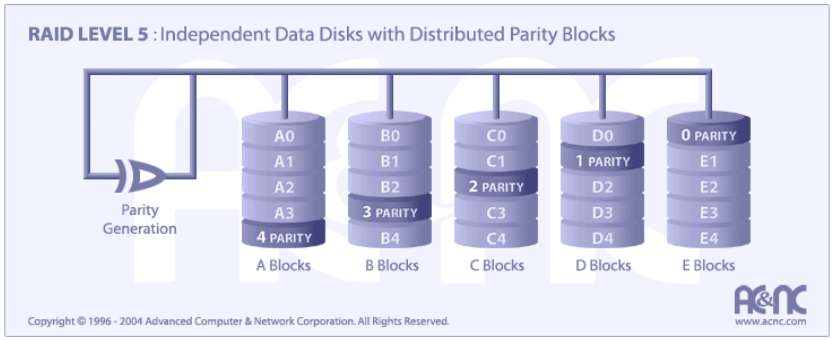
Sistemas RAID



Sistemas RAID



Sistemas RAID



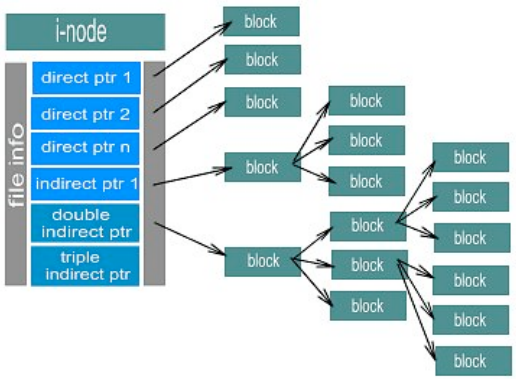
Estudo de casos

- MS-DOS
  - Baseado em FATs
    - File Allocation Tables indicam blocos ocupados por cada ficheiro e ainda os blocos livres na partição
    - Entrada na directoria indica o primeiro bloco do ficheiro. Para localizar o seguinte é preciso seguir a FAT
    - Dimensão da FAT ? Pode obrigar a overlays de partes da FAT
    - Duplicação de FATs para tolerar corrupção

Estudo de casos

- Unix
  - Directorias + I-nodes + data blocks
  - Directorias
    - São ficheiros especiais que fazem a associação nome / i-node
  - I-nodes contêm restantes atributos dos ficheiros, incluindo permissões (ugo), datas e localização dos blocos (até 3 níveis de indirecção)

Estudo de casos



Log-structured File Systems

- Devido à existência de caches em memória, e a necessidade de várias escritas em disco, há hipótese da informação ficar incoerente após crash => corrupção do SF
- FSCK pode demorar muito tempo pois tem de testar todos os meta-dados
  - inaceitável em certos cenários
- É preciso que o sistema de ficheiros recupere depressa

Solução?

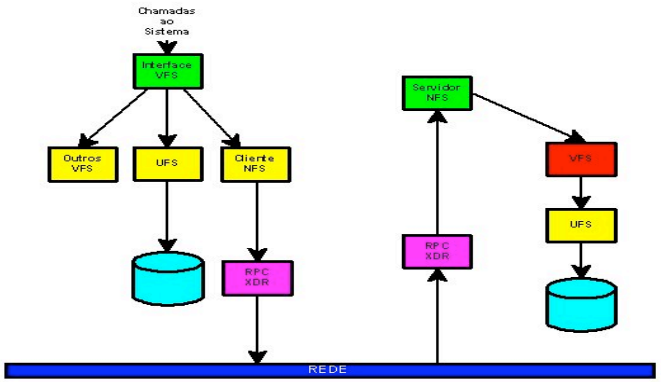
Log-structured File Systems

- A solução passa por utilizar as "boas práticas" dos sistemas de gestão de Bases de Dados...
- SGBDs há muito utilizam Logs para garantir as propriedades ACID (aqui interessa em particular a Atomicidade)
  - SGBD escrevem no Log operações e dados
  - FS tendem a escrever apenas meta-dados (i-nodes, free block allocation maps, i-nodes maps, etc.)

Log-structured File Systems

- Os sistemas de ficheiros baseados em "diário" (Log) mantêm um registo (log) das operações de actualização do SF.
  - As transacções são registadas no Log
  - Em background, as operações indicadas no Log são executadas sobre o sistema de ficheiros e a transacção marcada como committed. Em caso de crash, reexecuta-se apenas o log não completado
  - Checkpointing pode atrasar aplicações
  - Numa leitura, se o bloco pretendido não tiver sido alvo de "checkpoint" há que consultar o log => atraso.

Network File System (NFS)



E ainda...

- Extent-based file systems
- Parallel File Systems
- Distributed File Systems
- Storage Area Networks

...

O "estado-da-arte"

- Perquise no Google por Ext3, XFS, JFS, NTFS, Coda...
- Ou passe algum tempo em http://www.aspsys.com/software/links.aspx/14.aspx
- Se o tempo é limitado, recomenda-se a leitura de
  - Reiser FS (http://www.namesys.com/)

Backups

- Assegure-se que percebe a diferença entre
  - Backup
  - Redundância nos discos, por exemplo Raid-1(mirroring) ou Raid-5
- Backups
  - Para onde? Quando? Que garantias de integridade?

Programa

- Sistemas de ficheiros
  - Requisitos, objectivos, estudo de casos
  - RAID, Log struct FS, sistemas de ficheiros distribuídos
- Recapitulação de conceitos de programação concorrente
  - Modelos de concorrência, comunicação e sincronização (processos e threads)
- Mãos na massa:
  - Exercícios de programação concorrente com processos, semáforos, threads, mutexes…
  - Aulas práticas em laboratório: Linux

Porquê criar vários processos?

- Porque dá jeito... + conveniência
  - Estruturação dos programas
  - Para não estar à espera (spooling, background...)
  - Multiplas actividades / janelas
- Porque é melhor + eficiência
  - Múltiplos CPUs
  - Aumenta a utilização de recursos (e.g multiprogramação)