

Arquitetura de Computadores - PPGI/UEPB

(Unidade 6)

O ano de 1936 foi, sem dúvida, um marco para a ciência (não só da computação). Um jovem matemático britânico, conhecido como Alan Turing, publicava o primeiro artigo que apresentava ao mundo o seu modelo de máquina universal. Era uma “simples” máquina de computar, composta de um cabeçote capaz de mover-se por uma fita, na qual lia e escrevia informações. Estava pronta a base para, praticamente, toda a computação existente até hoje! De lá para cá, muita coisa mudou, mas quase tudo permaneceu o mesmo (isso mesmo, não é um erro de digitação). Os computadores mais modernos que utilizamos atualmente são basicamente implementações da máquina universal, proposta por Turing em 1936.

A fita dessa máquina, hoje em dia mais conhecida como memória, sempre se manteve como um foco de esforços por parte dos projetistas de hardware, que buscam aperfeiçoar suas capacidades de armazenamento, velocidade e consumo elétrico. Inicialmente usadas em meio magnético, logo as memórias voláteis passaram a ser elétricas, eliminando os gargalos mecânicos inconvenientes. Mas, claro, isso apenas não era suficiente. Logo os computadores passaram a usufruir de vários níveis de memória, as quais variam basicamente em velocidade. As mais velozes, como as memórias cache, naturalmente são mais caras e menos acessíveis. A divisão em níveis teve como objetivo otimizar

o acesso aos dados de memória, mas não provocando um grande impacto no custo. Por exemplo, ao invés de construir um computador apenas com memória cache como memória principal, constrói-se uma máquina com uma pequena parcela dessa memória (o que não trará grande impacto no custo), e aplicam-se mecanismos/algoritmos de gerenciamento, os quais buscarão garantir que os dados realmente úteis estejam, na maior parte do tempo, disponíveis na memória cache. Com isso, podemos simular uma memória de tamanho variável, apenas pelo gerenciamento da mesma.

A velocidade de acesso à memória principal não depende exclusivamente da sua própria tecnologia, mas, também, do barramento de dados de memória e do barramento de endereço de memória. Haja vista que a memória cache encontra-se praticamente dentro do processador, neste caso a velocidade dos barramentos não chega a ser um problema, o que, infelizmente, não é o caso das memórias *RAM. No entanto, nos últimos anos, a velocidade dos

Memórias - Tamanho ou Velocidade?

Escrito por Administrator

Sáb, 05 de Novembro de 2011 02:44 - Última atualização Sáb, 05 de Novembro de 2011 15:18

barramentos externos tem atingido valores mais altos, dando lugar à importância da velocidade interna do circuito de memória. Além disso, tais barramentos dobraram de tamanho, passando a ter 64 linhas (bits), não só no barramento de endereço, mas também no de dados. Com isso, ganhamos memórias muito mais velozes, além de apenas memórias de grande capacidade. É pena que, infelizmente, ainda é comum se ouvir propagandas de computadores com “muita memória” por preços relativamente muito baixos, enquanto outros computadores com “menos memória” (mas com uma tecnologia mais nova e muito mais veloz) por preços mais altos. Assim, o consumidor, mal influenciado, quase sempre termina fechando um belo mal negócio.

Terminando de arrumar as referências ainda... minha internet deu problema ;

(Resenha do Artigo:

FÁVERO, AL: “**Suporte a Multiprocessadores Simétricos (SMP) em kernel Linux**”,
Universidade Federal do Rio
Gande
do Sul.)

Memórias - Tamanho ou Velocidade?

Escrito por Administrator

Sáb, 05 de Novembro de 2011 02:44 - Última atualização Sáb, 05 de Novembro de 2011 15:18

Referências:

Open Source Development Labs, Inc., ***Linux Process Scheduler Improvements in Version***

2.6.0, Disponível em <http://developer.osdl.org/craiger/hackbench/>. Acesso em abril 2004.

Rick Lindsley, ***What's New in the 2.6 Scheduler***, Disponível em

<http://www.linuxjournal.com/article.php?sid=7178> . Acesso em abril 2004.

Enkh Tumenbayar, ***Linux SMP HOWTO***. Disponível em

<http://www.tldp.org/HOWTO/SMP-HOWTO.html>. Acesso em abril 2004.

Hank Dietz, ***Linux Parallel Processing HOWTO***, Disponível em

<http://www.tldp.org/HOWTO/Parallel-Processing-HOWTO.html>. Acesso em abril 2004.

Intel Corporation, ***MultiProcessor Specification, Version 1.4*** May 1997.