

## SCHIAVUX - SISTEMA OPERACIONAL LINUX PARA SIMULAÇÕES DE FÍSICA

**Autor: Luiz Eduardo Camargo Aranha Schiavo<sup>1</sup>, orientador: Dr. Irapuan Rodrigues de Oliveira Filho<sup>2</sup>**

<sup>1</sup>UNIVAP – Universidade do Vale do Paraíba Campus Platanus - FEAU - Faculdade de Engenharias, Arquitetura e Urbanismo - Av. Dr. Januário Miráglia, 3 - Vila Abernássia, 12460-000 - Campos do Jordão, São Paulo, [eduardo.ti@terra.com.br](mailto:eduardo.ti@terra.com.br)

<sup>2</sup>UNIVAP – Universidade do Vale do Paraíba – IP&D - Instituto de Pesquisa e Desenvolvimento - Av. Shishima Hifume, 2911 – Urbanova, 12244-000 – São José dos Campos, São Paulo, [irapuan@univap.br](mailto:irapuan@univap.br)

**Resumo-** Foi desenvolvida e implementada uma distribuição Linux com o intuito de realizar simulações computacionais de Física onde, no campo da Mecânica, são aplicados os conceitos de posição e deslocamento, movimentos uniformes, acelerados, circulares, lançamento vertical, queda livre, massa (como partícula), forças, princípio da conservação de energia, atrito e colisões. No campo da Gravitação são aplicados os conceitos de posição e deslocamento, massa, leis de Newton para gravitação e leis de Kepler. O sistema operacional foi compilado a partir do código fonte do Kernel x86, com diversas modificações e otimizações, visando o aumento de recursos e desempenho para simulações.

**Palavras-chave:** Linux, Física, Galáxia, C/C++, Kernel.

**Área do Conhecimento:** Engenharias, Ciências Exatas e da Terra.

### Introdução

Foi desenvolvido um sistema para simulações de Física (mecânica) [1], onde eram inseridos os valores de diversas forças atuando sobre uma ou mais partículas, o sistema decompunha em vetores [2] de 3 dimensões, calculava e demonstrava as resultantes. Todo sistema foi escrito em linguagem C [3] com alguns poucos trechos em linguagem C++ [4], sempre focando em respostas rápidas. Tendo o simulador de partículas funcionando o próximo passo foi implementar a simulação de galáxias[5].

A partir das dificuldades encontradas na tentativa de realizar tais simulações, aliadas à falta de recursos de processamento, a única solução encontrada foi modificar a estrutura da distribuição Linux [6] de forma a garantir os recursos necessários para fazer não somente as simulações iniciais, mas outras mais complexas.

Inicialmente com a distribuição Ubuntu 9.4 [7], foram removidos diversos serviços desnecessários como codec de áudio e vídeo, serviço de impressão, Bluetooth, área de trabalho remota entre outros, mas a diferença no desempenho não foi significativa. A partir desse ponto ficou claro que para uma melhora no desempenho a distribuição deveria ser compilada a partir do código fonte.

Neste ponto foi escolhida a distribuição LFS – Linux from scratch 6.8 [8] que permitiu uma série de personalizações que antes não seriam possíveis. Após dias compilando módulos e dependências, já foi possível notar um primeiro

ganho significativo de desempenho, mas também se observou a necessidade de se aprofundar mais e compilar o Kernel [9] do Linux [6] e foi aqui onde realmente todo esse trabalho começou.

### Material e Métodos

#### O Sistema Operacional:

Para iniciar o trabalho foram definidos alguns padrões para facilitar qualquer reuso de código no futuro.

- Unix-like

O padrão de Sistema Operacional adotado foi o Unix-like [10], Sistema Operacional tipo Unix, ou seja, um Sistema Operacional parecido com Unix, mas que não segue a norma paga SUS - Single Unix Specification [10].

- Kernel monolítico

A arquitetura de Kernel [9] adotada foi monolítica [11], ou seja, todo o Kernel é executado em uma única área de execução com supervisão, garantindo maior desempenho, mas maior margem para erros de programação com memória, ao contrário que teria conseguido nos modelos mais novos e mais seguros adotados pelo Mac OS X e Windows.

- Linux x86 2.6.37

O Kernel [9] adotado foi o 2.6.37 de 32 bits, permitindo que o sistema operacional seja executado em qualquer computador pessoal.

**- Versões do código fonte**

Bash 4.2, Binutils 2.21, Coreutils 8.10, GCC 4.5.2, Glibc 2.13, Grep 2.7, Linux 2.6.37 [6], Make 3.82, X-Window X11R7.6 [12] e as respectivas dependências.

**- Linguagens e técnicas usadas**

A grande maioria dos códigos é escrita em C [3] (clássico), com poucos trechos em C++ [4], um dos pontos mais fortes de ambas as linguagens que foi amplamente usado é o recurso ponteiros, sendo usado tanto para alocação dinâmica de memória como para passagem de valores por referência, ambos os usos são extremamente úteis para ganho de desempenho, pois reduzem os “passos” no acesso aos dados na memória.

Uma das maiores vantagens de usar a alocação dinâmica de memória é também a maior desvantagem, a velocidade ganha em relação a matrizes e vetores [13] convencionais é enorme, mas em contrapartida o código pode se tornar confuso ocasionando erros. Em linguagens mais novas esse recurso de acesso direto a memória não é mais permitido.

Outro ponto importante para desempenho é que foi evitado ao máximo o uso de recursos como sobrecarga, polimorfismo e classes [14], usando apenas quando são extremamente necessários, pois apesar de práticos, esses recursos consomem mais passos e tempo de processamento, além de gerar um binário mais “pesado”.

**- Equipamento usado**

O equipamento usado foi um PC Intel Pentium 4 3.2 Ghz com 2 threads e 1 GB de memória DDR 400, 2 HD's SATA de 80 GB Seagate 7200RPM, rodando Schiavux [15] e Ubuntu 9.4 [7] alternadamente.

**- Compilação**

A Compilação foi feita em 2 estágios. Primeiro foi usado um sistema Ubuntu 9.4 [7] para compilar o sistema “live limpo” que serviria de base para compilar o sistema definitivo, esse primeiro processo gerou um sistema funcional carregado apenas na memória e com o único intuito de compilar o próximo estágio.

Tendo o sistema base “live limpo”, compilado e “rodando” o segundo estágio foi compilar novamente todo o sistema, mas desta vez movendo os binários para um HD vazio que seria o sistema definitivo.

A compilação em 2 estágios permitiu um sistema “limpo” sem bibliotecas desnecessárias usadas apenas na compilação e instalação inicial. Esse processo, apesar de lento, levou a um aproveitamento melhor de recursos, tendo em vista que bibliotecas que antes eram apenas

usadas na compilação e linkedição não precisam ficar residentes na memória.

**Pontos fortes do sistema:**

Como o sistema operacional ficou livre de sistemas desnecessários, o tempo de inicialização foi muito reduzido, de 32 para 22 segundos.

O ganho de rendimento em simulações também foi expressivo, rodando a mesma simulação o Schiavux [15] foi 17% mais rápido do que no Ubuntu 9.4 [7].

**Simulação de Forças Físicas:**

Para facilitar a implementação foi utilizada a decomposição das forças [1] em vetores [2] de três dimensões, calculando os devidos movimentos de cada dimensão separadamente, ou seja, o movimento é tratado como unidimensional, mas as resultantes [1] são apresentadas em até 3 dimensões.

Unidades: as unidades de apresentação para todos os resultados são dadas nos padrões do SI [1] (Sistema Internacional de Unidades), quando necessário as unidades são apresentadas em notação científica (valores muito grandes e muito pequenos).

Todas as grandezas físicas calculadas são armazenadas em vetores [3] de ponto flutuante, onde o índice de acesso é o tempo.

Limites adotados: para evitar o estouro de variáveis, o tempo máximo para cada movimento é de 1 semana, a distância máxima de 1 milhão de quilômetros.

Os conceitos físicos implementados foram[1]:

- Movimento em uma, duas e três dimensões
- Posição e deslocamento.
- Velocidade média e instantânea.
- Aceleração média e instantânea.
- Queda Livre e Lançamento Vertical.
- Projeteis.
- Movimento Circular.
- Forças.
- Massa (em partícula).
- Forças de Atrito.
- Arrasto e Velocidade Terminal.
- Trabalho e Energia Cinética.
- Força Elástica.
- Colisões.

**Simulação de Galáxias:**

Para facilitar a implementação foi utilizada a decomposição das forças gravitacionais [16] em vetores [2] de três dimensões, calculando os devidos movimentos de cada corpo (estrela, satélite ou planeta) [5] em dimensões separadas, ou seja, o movimento é tratado como

unidimensional, mas as resultantes [1] são apresentadas em 3 dimensões.

Unidades: as unidades de apresentação para todos os resultados são dadas nos padrões de Unidade Astronômica [16] (ano luz e parsec), como as grandezas são de tamanho considerável, as unidades em alguns casos são apresentadas em notação científica.

Todas as grandezas físicas calculadas são armazenadas em vetores [3] de ponto flutuante, onde o índice de acesso é o tempo.

Limites adotados: para evitar o estouro de variáveis, o tempo máximo para cada simulação é de 900 mil anos, a distância máxima de 1 milhão de parsec.

Para simulação mais realista, foi feita a distribuição homogênea [5] de corpos nas galáxias [5] simuladas.

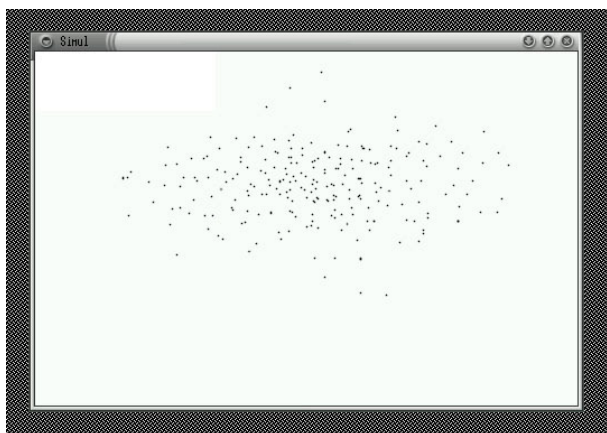
Os conceitos implementados foram [16]:

- Movimento em uma, duas e três dimensões.
- Posição e deslocamento.
- Velocidade média e instantânea.
- Aceleração média e instantânea.
- Massa (como partícula).
- Lei da gravitação de Newton.
- Leis de Kepler.
- Colisão de Galáxias.

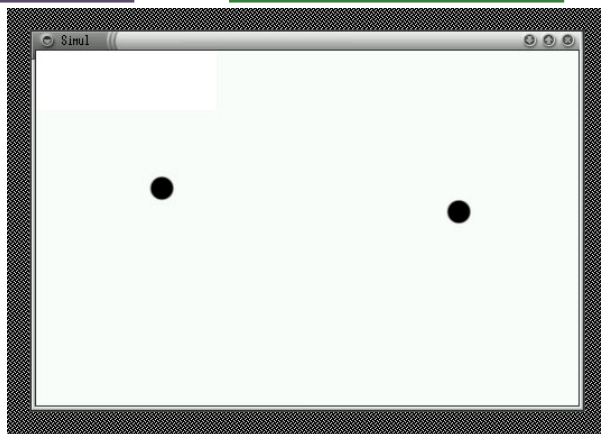
## Resultados

O resultado obtido foi uma distribuição Linux [6] sólida e estável derivada do Kernel [9] monolítico [11] x86 2.6.37, com as funcionalidades de um Sistema Operacional Unix-like [10] e aplicações voltadas para simulações computacionais de Física que foi chamado de SCHIAVUX [15].

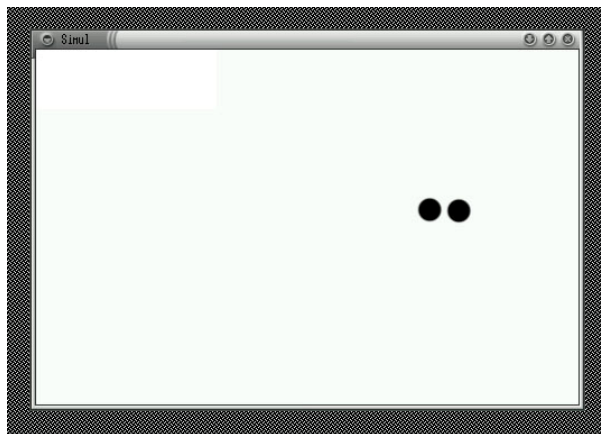
Alguns exemplos das telas de simulação



Tela de simulação de galáxia.



Tela de simulação de forças físicas colisão de partículas.



Tela de simulação de forças físicas colisão de partículas.

## Discussão

Os resultados encontrados foram mais que satisfatórios, pois com a criação desta distribuição Linux [6] foi possível realizar simulações que foram cerca de 17% mais rápidas que em distribuições convencionais. Dado o resultado favorável, é encorajadora a implementação de simulações com maiores graus de dificuldade para as próximas edições da INIC, como por exemplo, Transferência de Calor e Mecânica de Fluidos.

## Conclusão

A implementação a partir do código fonte do sistema operacional Linux [6], apesar de extremamente trabalhosa, é também muito vantajosa e deve ser encorajada para uso em pesquisa e simulações computacionais de alto desempenho em ambiente Unix-like [10].

**Agradecimentos:**

Em primeiro lugar a minha esposa Laura Domingues Schiavo, pelas noites de sono perdidas, aos meus pais Antonio Luiz Schiavo Junior e Hosana Camargo Schiavo, que me incentivaram sempre.

Meus irmãos, Carlos Eduardo Camargo Aranha Schiavo e Juliana Camargo Aranha Schiavo que me apoiaram e incentivaram.

Em especial ao meu irmão Luiz Augusto Camargo Aranha Schiavo que me apresentou ao ramo de pesquisa.

Ao professor, orientador e amigo Dr. Irapuan Rodrigues de Oliveira Filho que me aguentou com toda minha falta de tempo e me mostrou os caminhos da pesquisa.

Aos professores, Dr. Rodrigo Sávio Pessoa, Dra. Diana Paula Andrade Pilling Guapyassú de Oliveira e Dra. Alessandra Abe Pacini que me incentivaram a embarcar nessa jornada.

**Referências**

- [1] - HALLIDAY, David. Fundamentos de fisica - v. 1: mecânica. 8. ed. Rio de Janeiro, RJ: LTC, 2009.
- [2] - STAINBRUCH, Alfredo. Geometria Analítica - 2. Ed, São Paulo, SP, MacGraw-Hill, 1987.
- [3] - MIZRAHI, Victorine Viviane. Treinamento em linguagem c: modulo 1. São Paulo: Pearson Education do Brasil, 1995.
- [4] - MIZRAHI, Victorine Viviane. Treinamento em linguagem c++: modulo 1. São Paulo: Pearson Makron Books, 1995.
- [5] - BERTIN, G. Dynamics of galaxies. Cambridge: Cambridge University Press, 2000. Xiv.
- [6] - BACH, Maurice J. The design of the unix operating system. Englewood Cliffs: Prentice Hall, 1986 (Prentice Hall International Editions).
- [7] - Ubuntu Linux - <http://www.ubuntu.com/> acessado em 28/08/2011.
- [8] - BEEKMANS, Gerard. Linux From Scratch, V 6.8, <http://www.linuxfromscratch.org/lfs/downloads/stable/LFS-BOOK-6.8.pdf> acessado em 21/8/2011.
- [9] - The Linux Kernel Archives <http://www.kernel.org/> acessado em 21/8/2011.
- [10] - The Open Group - <http://www.unix.org/version4/> acessado em 21/8/2011.
- [11] - The Linux Information Project - <http://www.linfo.org/kernel.html> acessado em 28/8/2011.
- [12] - X.Org Project <http://www.x.org/wiki/Releases/7.6> acessado em 28/8/2011.
- [13] - MIZRAHI, Victorine Viviane. Treinamento em linguagem c: modulo 2. São Paulo: Pearson Education do Brasil, 1995.
- [14] - MIZRAHI, Victorine Viviane. Treinamento em linguagem c++: modulo 2. São Paulo: Pearson Makron Books, 1995.
- [15] - Schiavux, Site do projeto, <http://www.eduschiavo.com/schiavux> acessado em 21/8/2011.
- [16] - HALLIDAY, David. Fundamentos de fisica - v. 2: gravitação, ondas e termodinâmica. 8. ed. Rio de Janeiro, RJ: LTC, 2009.