

Rafael Coninck Teigão
Julio Henrique Morimoto

Estudo de Viabilidade
BananaKernel: Um Sistema Operacional
Didático

Curitiba - Paraná

Março 2004

Rafael Coninck Teigão
Julio Henrique Morimoto

Estudo de Viabilidade
BananaKernel: *Um Sistema Operacional*
Didático

Trabalho apresentado pelos alunos do 7º período do curso de Bacharelado em Ciência da Computação para a disciplina Projeto Final I.

Orientador:
Prof. Dr. Carlos Alberto Maziero

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

Curitiba - Paraná

Março 2004

Sumário

Lista de Figuras	p. iii
Lista de Tabelas	p. iv
Lista de Acrônimos Utilizados neste Documento	p. v
1 Objetivo	p. 1
2 Escopo	p. 2
3 Diagnóstico Atual	p. 4
4 Alternativas Propostas	p. 5
4.1 Alternativa “Nachos”	p. 5
4.2 Alternativa “Zero”	p. 6
4.3 Alternativa “Modificação”	p. 6
4.4 Alternativa “OSKit”	p. 6
5 Alternativa Recomendada	p. 8
6 Benefícios	p. 9
6.1 Benefícios tangíveis	p. 9
6.2 Benefícios intangíveis	p. 9
7 Custos	p. 10
7.1 Custos de implementação	p. 10

7.2 Custos de implantação	p. 10
8 Riscos	p. 11
9 Cronograma	p. 14
10 Conclusão	p. 15
Referências Bibliográficas	p. 16
Índice Remissivo	p. 17

Lista de Figuras

1	Definição do Escopo do Projeto	p. 2
2	Componentes disponibilizados pelo OSKit	p. 7
3	Cronograma - Etapas do Desenvolvimento da Solução	p. 14

Lista de Tabelas

1	Custos de Implementação	p. 10
2	Custos de Implantação	p. 10
3	Nomenclatura dos valores de Risco	p. 11
4	Quantificação dos Riscos	p. 11
5	Riscos X Percentagens	p. 13

Lista de Acrônimos Utilizados neste Documento

OS - *Operating System* - Sistema Operacional, p. 1

PC - *Personal Computer* - Computador Pessoal, p. 5

MM - *Memory Management* - Gerência de Memória, p. 6

FS - *File System* - Sistema de Arquivos, p. 6

MHz - *Megahertz*, p. 10

MB - *Megabyte*, p. 10

RAM - *Random Access Memory* - Memória de Acesso Randômico, p. 10

HD - *Hard Disk* - Disco Rígido, p. 10

GB - *Gigabyte*, p. 10

1 *Objetivo*

O propósito do sistema é proporcionar uma ferramenta para auxiliar no ensino da disciplina Sistemas Operacionais, demonstrando alguns dos mecanismos internos de funcionamento de um *OS*¹ moderno.

O sistema deve apresentar mensagens explicando partes do funcionamento, deve possuir uma documentação e um desenvolvimento que possibilite aos alunos e professores a criação de novos módulos para substituir ou acrescentar funcionalidades.

¹*Operating System* - Sistema Operacional

2 *Escopo*

Existe uma necessidade no meio acadêmico de um sistema que facilite o ensino de conceitos de OS para alunos, principalmente dos cursos de Bacharelado em Ciência da Computação e Engenharia da Computação.

Este sistema deve possuir uma documentação que explique o funcionamento de conceitos importantes, como gerência de arquivos e memória, por exemplo, e ainda permita que o aluno ou o professor faça substituições em módulos responsáveis pela apresentação desses conceitos.

O escopo deste projeto, demonstrado na figura 1, cobre a criação de um sistema operacional simples, contendo módulos de gerência de memória, gerência de arquivos e escalonador de processos, sendo que esses módulos devem apresentar mensagens na tela do computador explicando o funcionamento interno de cada subsistema.



Figura 1: Definição do Escopo do Projeto

O projeto cobre, também, a criação de um manual de usuário que explique como incluir ou substituir módulos, além de uma breve introdução à tecnologia e ao método de desenvolvimento utilizado na criação do sistema.

Este projeto não cobre, porém, ensinar aos usuários linguagens de programação, nem pretende montar um curso de sistemas operacionais, como mostra a figura 1. O fornecimento do equipamento necessário para implantação, as demais ferramentas para a execução do sistema ou para condução das aulas e os *softwares* e sistemas necessários para o desenvolvimento ou modificação de módulos também não estão cobertos.

Assume-se, como premissas, que os usuários têm conhecimento e capacidade de criar programas nas linguagens de programação adequadas à alternativa escolhida, bem como capacidade de compreender textos escritos em inglês técnico, além de estarem sendo acompanhados por um professor capacitado da disciplina Sistemas Operacionais.

3 *Diagnóstico Atual*

Muitos professores de Sistemas Operacionais recorrem a implementações para demonstrar alguns dos conceitos aprendidos em sala. Porém, essa forma de aprendizado encontra vários problemas, entre eles:

- falta de uma documentação clara que auxilie os alunos;
- falta de um ambiente dedicado para a implementação, o que obriga os alunos a lidarem com sistemas completos e muito mais complexos;
- muitas vezes o OS base não fornece mensagens suficientemente didáticas de erro e funcionamento ao aluno;
- dificuldades para resolver problemas, uma vez que não se tem controle do OS em que se implementa.

Com isso, o desenvolvimento de uma simples biblioteca de *threads*, por exemplo, demanda muito tempo dos alunos, torna-se uma tarefa extremamente dependente do OS base, e fornece conhecimento apenas superficial do processo.

Um OS didático, que permita fácil troca de módulos, bem documentado, aceleraria este processo de desenvolvimento, ao fornecer um ambiente de desenvolvimento propício; melhoraria o controle sobre os módulos, pois a complexidade do OS base seria muito menor; e auxiliaria no aprendizado do aluno com informações úteis na tela.

4 *Alternativas Propostas*

A seguir serão propostas quatro alternativas, com suas vantagens e desvantagens descritas brevemente. A primeira alternativa é um sistema já existente.

As três alternativas por nós abaixo propostas permitem inserção ou alteração de funcionalidades por parte dos usuários, e fornecem informações sobre o funcionamento interno na tela do computador.

4.1 Alternativa “Nachos”

Como encontrado em [Anderson 1996], o Nachos é um *software* para ensinar alunos de graduação em cursos de Sistemas Operacionais. O Nachos é composto por:

- uma documentação geral;
- um código-base de um OS funcional;
- um simulador de um *PC*¹ genérico;
- exercícios de exemplo; e
- uma introdução à linguagem C++.

Apesar do Nachos ser um sistema maduro e ser acompanhado de uma documentação, seu tamanho e complexidade são fortes pontos negativos. Além disso, não apresenta mensagens didáticas durante a execução do OS, seus comentários estão todos em inglês, e é codificado usando a linguagem C++, sendo que a ênfase em cursos de Ciência da Computação está na linguagem C.

A licença de uso do Nachos é gratuita, desde que sejam mantidos os avisos de *Copyright*.

¹*Personal Computer* - Computador Pessoal

O Nachos pode ser encontrado no seu endereço de Internet² e maiores informações podem ser conseguidas com o autor pelo seu endereço de *e-mail*³.

4.2 Alternativa “Zero”

Esta alternativa consiste em criar, sem qualquer base de código de apoio, um sistema operacional que possa ser carregado, lançar um gerenciador de memória e de arquivos e um escalonador de processos.

A principal vantagem desta alternativa é o controle total das funções dentro do sistema, uma vez que todo o código necessário seria desenvolvido por nossa equipe.

A desvantagem mais forte é o grande tempo necessário para tal desenvolvimento. Seria certamente necessário um período superior a um ano de desenvolvimento para se obter um sistema funcional.

4.3 Alternativa “Modificação”

Esta alternativa demanda menor tempo de desenvolvimento e maior tempo de criação da documentação.

Parte-se de um OS simples, como o PicoBSD [Nair 1998], ou o Trinux [Franz 2003], e cria-se as mensagens didáticas sobre as partes já prontas do sistema.

Com isso, tem-se menor controle sobre o *software*, porém é possível obter um sistema funcional rapidamente.

4.4 Alternativa “OSKit”

Esta alternativa visa utilizar o conjunto de bibliotecas conhecido como Flux OSKit [Ford et al. 1997] para desenvolver o sistema.

As bibliotecas fornecem diversas funcionalidades, totalizando 34 bibliotecas componentes, como mostra a figura 2, que possibilitam modificações e inclusões rápidas de novos recursos ao OS (*e.g.* *MM*⁴, *FS*⁵ e gerência de processos).

²<http://www.cs.washington.edu/homes/tom/nachos/>

³nachos@cs.berkeley.edu

⁴*Memory Management* - Gerência de Memória

⁵*File System* - Sistema de Arquivos

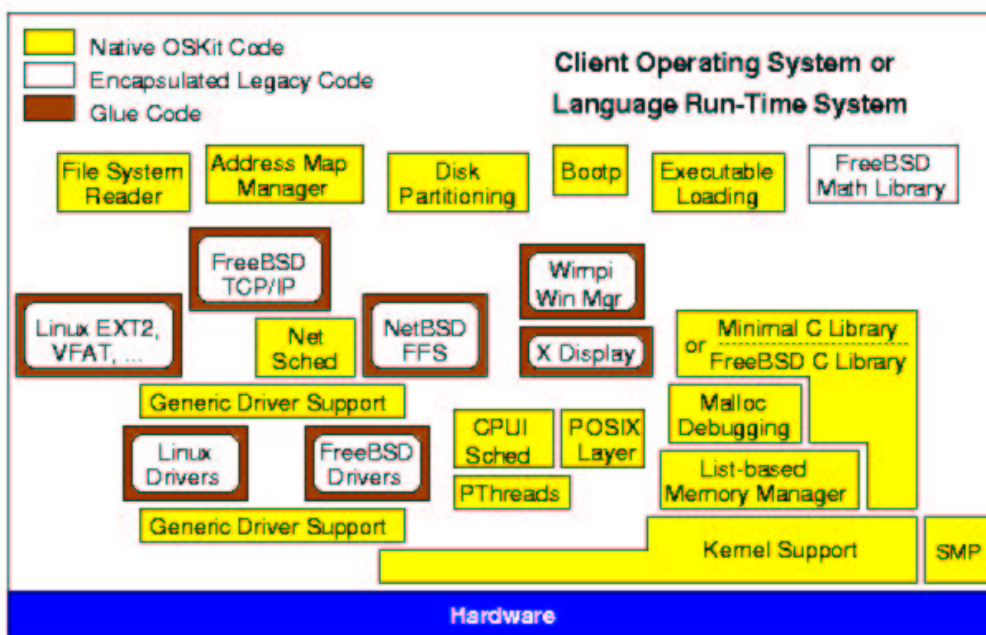


Figura 2: Componentes disponibilizados pelo OSKit

Os documentos fornecidos pela equipe do OSKit [The Flux Research Group 2002] também podem servir como documentação para o aluno ou professor, uma vez que qualquer componente criado com o OSKit poderá, com pequenas modificações, ser incluído no OS final.

5 *Alternativa Recomendada*

A alternativa recomendada é a “OSKit”, citada no capítulo 4.4, pelos seguintes motivos:

1. permite um desenvolvimento mais rápido e ordenado do OS;
2. o OSKit facilita a criação de novos módulos, o que poderia ajudar nas aulas;
3. o OSKit é bem documentado;
4. outros desenvolvedores poderiam facilmente acrescentar funcionalidades utilizando as bibliotecas do OSKit;
5. é uma opção mais simples para o desenvolvimento de um sistema modular; e
6. partes do sistema que não possuam um grande interesse acadêmico podem ser diretamente importadas do OSKit.

Como será visto no capítulo 8, a alternativa “Modificação”, capítulo 4.3, poderá ser considerada, em caso de falha ou problemas com a alternativa “OSKit”, porém, com perdas aceitáveis na qualidade final do sistema.

6 *Benefícios*

Os benefícios foram divididos em tangíveis, que podem ser quantificados, e intangíveis, que podem não ser claramente visíveis por não serem facilmente quantificáveis.

6.1 Benefícios tangíveis

Os benefícios tangíveis levam em consideração os custos (capítulo 7).

- As ferramentas de *software* utilizadas na implementação e na implantação são gratuitas.
- A licença escolhida para o projeto foi uma licença de código aberto e o sistema final é gratuito.
- Considerando os dois aspectos anteriores, pode-se utilizar o sistema, em uma infraestrutura de equipamentos já existente, a um custo zero de *software*.

6.2 Benefícios intangíveis

- Professores terão uma ferramenta que irá auxiliar nas aulas de Sistemas Operacionais.
- Alunos terão maior facilidade em visualizar o funcionamento de um OS.
- O sistema final e o relato de experiências desenvolvidas podem ajudar a divulgar, pelo meio acadêmico, os nomes dos desenvolvedores.

7 Custos

Os custos foram divididos em custos de implementação, que serão inicialmente pagos pela equipe de desenvolvimento, e custos de implantação, que serão pagos diretamente pelos usuários.

7.1 Custos de implementação

Os custos de implementação foram agrupados na tabela 1.

Tabela 1: Custos de Implementação

Descrição	Preço Unitário	Quantidade	Preço Total
Biblioteca OSKit e <i>software</i> de desenvolvimento	Gratuito	2	R\$ 0,00
Computador AMD K6-2 500 MHz, 128 MB RAM, HD 20 GB (já disponível à equipe)	R\$ 1.030,00	2	R\$ 2.060,00
		Total	R\$ 2.060,00

7.2 Custos de implantação

Os custos de implantação foram agrupados na tabela 2, e correspondem aos custos por aluno.

Tabela 2: Custos de Implantação

Descrição	Preço Unitário	Quantidade	Preço Total
Biblioteca OSKit e <i>software</i> de desenvolvimento (não coberto pelo escopo)	Gratuito	1	R\$ 0,00
Computador AMD K6-2 500MHz, 32MB RAM, HD 20GB (não coberto pelo escopo)	R\$ 600,00	1	R\$ 600,00
		Total	R\$ 600,00

8 *Riscos*

O quadro de riscos foi efetuado com base na metodologia de probabilidade e impacto dos riscos envolvidos no projeto. Para calcular um fator de risco para o projeto, foram adotados valores de grandeza para cada risco, tornando tangíveis a probabilidade do risco vir a ocorrer e o impacto (peso) que o mesmo causaria no andamento do projeto. A nomenclatura dos valores utilizados encontra-se na tabela 3.

Tabela 3: Nomenclatura dos valores de Risco

Valor Associado	Significado
1	Muito baixo
2	Baixo
3	Médio
4	Alto
5	Muito alto

Dentro desta nomenclatura, podemos definir cada risco associado ao projeto, como na tabela 4. O risco total do projeto é dado pela média ponderada dos riscos. A média ponderada é calculada pelo somatório da probabilidade de ocorrência, multiplicada pelo impacto do risco de cada item relacionado na tabela 4, dividido pelo somatório do impacto do risco destes itens, ou $\frac{\sum P*I}{\sum I}$, em que P é a probabilidade de ocorrência e I é o impacto do risco.

Tabela 4: Quantificação dos Riscos

Risco	P	I	Explicação	Medida Preventiva	Medida Corretiva
Estruturação do Projeto					
Impossibilidade de prever quais as necessidades do projeto	1	3	Não prever todas as necessidades pode levar a muitas mudanças	Melhorar levantamentos de requisitos	Verificar com professores de OS as necessidades
Continua...					

Tabela 4: Quantificação dos Riscos - Continuação

Risco	P	I	Explicação	Medida Preventiva	Medida Corretiva
Os requisitos do projeto mudam constantemente ao longo do seu ciclo de vida	1	3	Mudanças bruscas nos requisitos podem aumentar o custo e o tempo de execução	Ter um escopo claro e bom levantamento de requisitos	Marcar reuniões freqüentes de levantamento e validação de requisitos com o orientador
Tecnologia do Projeto					
A equipe de desenvolvimento não domina a tecnologia utilizada	2	3	A falta de domínio da tecnologia pode atrasar, ou impossibilitar a execução do projeto	Levantar documentação necessária	Caso seja impossível dominá-la, abortar e usar a alternativa "Modificação"
Interfaces com este projeto					
Existem projetos em paralelo que podem afetar o desenvolvimento deste projeto	1	1	Projetos em paralelo podem modificar as ferramentas necessárias para a execução	Manter-se atualizado com as listas de discussão e de notificação de mudanças das ferramentas	Conseguir, com os desenvolvedores terceiros, a lista de modificações e preparar o projeto para elas
Cronograma					
O prazo de entrega não está sob controle	2	4	O prazo pode não ser suficiente	Verificar antecipadamente os prazos	Procurar com os responsáveis um adiamento; mudar para a alternativa "Modificação"
Continua...					

Tabela 4: Quantificação dos Riscos - Continuação

Risco	P	I	Explicação	Medida Preventiva	Medida Corretiva
A equipe de desenvolvimento não está segura quanto ao prazo do projeto	1	3	O prazo pode não ser suficiente	Verificar se o prazo está de acordo com o tempo esperado de desenvolvimento	Rever os prazos. Se não for possível adiamento, mudar para a alternativa “Modificação”
Competência da equipe					
A equipe de desenvolvimento não possui experiência prévia neste tipo de projeto	1	2	A falta de experiência da equipe pode levar a um mal entendimento técnico do projeto	Garantir que um orientador experiente supervisione o projeto	Procurar ajuda de um co-orientador com a experiência necessária
A equipe de desenvolvimento não possui treinamento nas ferramentas utilizadas	1	2	A falta de treinamento pode interferir no planejamento e execução	Levantar os documentos necessários	Procurar ajuda em listas de discussão sobre as ferramentas
Risco total do projeto (média ponderada)					1,334

Com base na metodologia de análise de risco adotada, conclui-se que o risco do projeto é suficientemente baixo, como mostra a tabela 5.

Tabela 5: Riscos X Percentagens

Valor Associado	Faixa de Percentagem	Significado
1	0% - 20%	Muito baixo
2	20% - 40%	Baixo
3	40% - 60%	Médio
4	60% - 80%	Alto
5	80% - 100%	Muito alto

O valor total (1,334) posiciona o projeto na faixa de risco entre 0% e 20%.

9 Cronograma

A figura 3 mostra a lista das etapas do desenvolvimento da solução.

Atividade	Data de início	Data de fim	Duração dias	Total dias acumulado
Levantamento de requisitos	23/03/2004	02/04/2004	8	8
Modelo comportamental	05/04/2004	07/05/2004	25	33
<i>DFD Individual</i>	<i>05/04/2004</i>	<i>12/04/2004</i>	<i>5</i>	<i>-</i>
<i>DFD Detalhado</i>	<i>12/04/2004</i>	<i>16/04/2004</i>	<i>5</i>	<i>-</i>
<i>Dicionário de dados</i>	<i>19/04/2004</i>	<i>23/04/2004</i>	<i>5</i>	<i>-</i>
<i>Diagrama de transição de estados</i>	<i>26/04/2004</i>	<i>30/04/2004</i>	<i>5</i>	<i>-</i>
<i>Especificação de processos</i>	<i>03/05/2004</i>	<i>07/05/2004</i>	<i>5</i>	<i>-</i>
Protótipo	10/05/2004	14/05/2004	5	38
Plano de testes	17/05/2004	20/05/2004	5	43
Conclusões	24/05/2004	27/05/2004	4	47
Total				47

Figura 3: Cronograma - Etapas do Desenvolvimento da Solução

A etapa de implementação ainda não está individualmente definida, mas a data para a entrega do projeto completo (códigos-fonte, documentação e manual do usuário) é 12 de novembro de 2004.

O cronograma será atualizado quando for efetuado o Projeto Lógico.

10 Conclusão

A execução deste estudo de viabilidade possibilitou destacar os seguintes pontos com relação ao projeto:

- existem vários benefícios para alunos e professores de Sistemas Operacionais;
- os custos de implementação são baixos;
- os custos de implantação são baixos; e
- o risco de falha do projeto é considerado baixo.

Acreditando que os benefícios para o meio acadêmico, tangíveis e intangíveis, sejam suficientes, e considerando os baixos custos e riscos, este projeto é considerado viável.

Curitiba, 25 de março de 2004.

Alunos responsáveis:

Rafael Coninck Teigão

Julio Henrique Morimoto

Referências Bibliográficas

- [Anderson 1996]ANDERSON, T. *README for the Nachos Release*. <http://www.cs.washington.edu/homes/tom/nachos/README> visitado em 15 de Marco de 2004, 1996.
- [Ford et al. 1997]FORD, B. et al. The Flux OSKit: A substrate for OS and language research. In: *Proceedings of the 16th ACM Symposium on Operating Systems Principles*. Saint-Malo, France: [s.n.], 1997.
- [Franz 2003]FRANZ, M. *Trinux Documentation*. <http://trinux.sourceforge.net/docs.txt> visitado em 15 de Marco de 2004, July 2003.
- [Nair 1998]NAIR, D. *The PicoBSD FAQ*. <http://people.freebsd.org/picobsd/picobsd225/faq.html> visitado em 15 de Marco de 2004, 1998.
- [The Flux Research Group 2002]The Flux Research Group. *The OSKit: The Flux Operating System Toolkit*. Version 0.97 (snapshot 20020317). Salt Lake City, UT, USA 84112, March 2002.

Índice Remissivo

Benefícios
 intangíveis, 9
 tangíveis, 9

Custos
 de implantação, 10
 de implementação, 10

licença, 5, 9, 10

Nachos, 5

OSKit, 6, 8, 10

Projeto Lógico, 14

Riscos, 11

Sistema Operacional
 de um disco
 PicoBSD, 6
 Trinux, 6
 Nachos, 5