



Instituto de Ciências Matemáticas e de Computação

Trabalho de SCE 217 – Programação Concorrente

Componentes de um Escalonador de Processos

Fabio Yoneyama

Gustavo Ribeiro Amigo

Eduardo Henrique Dias Aranega

Jorge David Novaes Pimentel Junior

SUMÁRIO

- 1) Introdução
 - 1.1 Conceitos Básicos
 - 1.2 Algoritmos de Escalonamento
 - 1.3 Computação Paralela e Sistemas Distribuídos

- 2) Escalonamento de Processos em Sistemas Distribuídos
 - 2.1 Componentes de um Escalonador
 - 2.2 Escalonamento Centralizado
 - 2.3 Escalonamento Distribuído

- 3) Conclusões

- 4) Referências Bibliográficas

1) INTRODUÇÃO

Esta monografia irá apresentar conceitos sobre o *Escalonador de Processos* e os respectivos componentes envolvidos na tarefa de alocação de processos. O trabalho do escalonador é determinar qual o processo vai rodar a seguir, levando em conta vários fatores como o tempo de resposta, a eficiência e a justiça. Iremos apresentar um estudo sobre escalonamento, traçando conceitos básicos.

1.1 – Conceitos Básicos

O escalonamento consiste em atribuir processos a processadores e determinar em que ordem estes processos serão executados. Ele é de extrema importância para sistemas paralelos e distribuídos, podendo ser considerado um dos problemas mais desafiantes nesta área.

Segundo Casavant, o escalonamento pode ser visto como um recurso para gerenciar recursos. Este gerenciador é basicamente um mecanismo ou uma política usada para controlar o acesso e o uso de vários recursos por seus vários consumidores.

Duas propriedades que devem ser consideradas no escalonamento: a satisfação do consumidor em relação a como o escalonador manipula os recursos em questão (desempenho); e a satisfação do consumidor em relação à dificuldade ou custo de acesso ao gerenciamento de recursos em si (eficiência).

Existem vários tipos de escalonamentos e algumas taxonomias foram propostas, porém nenhuma destas taxonomias abrange todas as características que um escalonador pode possuir; devido a isso existem características que serão descritas a parte. Dentre as taxonomias estudadas, a proposta por Casavant é a mais aceita e completa. A figura 1 mostra a taxonomia de escalonamento proposta pelo mesmo.

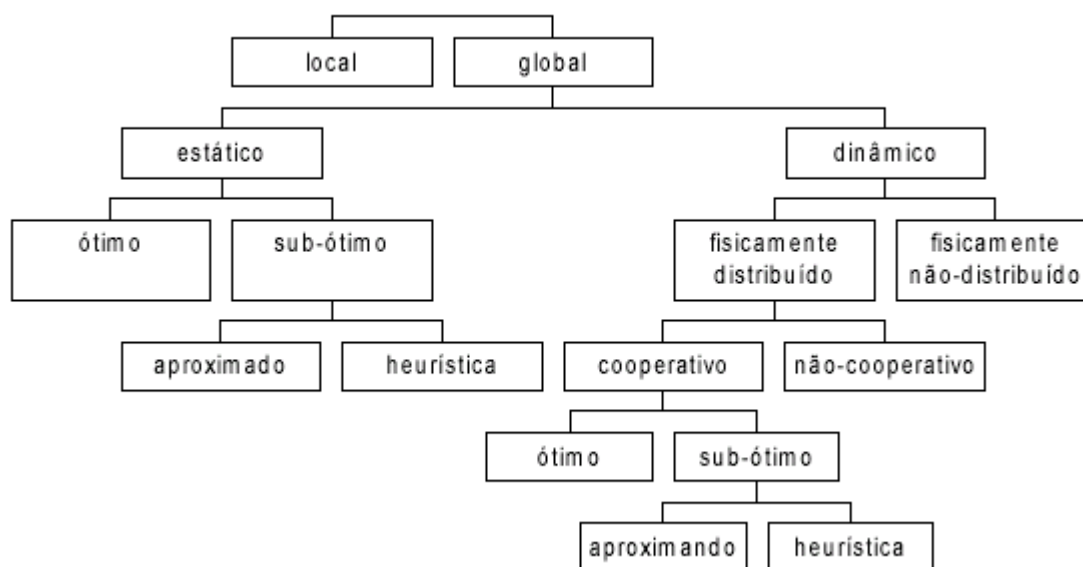


Figura 1 – Taxonomia de Escalonamento de Casavant

Na taxonomia apresentada o escalonamento local está relacionado com a ordenação de processos em um único processador.

O escalonamento estático é aquele que é resolvido antes da execução das tarefas de maneira determinística. Quando é possível determinar todas as informações destas tarefas (dependência entre tarefas, custo de comunicação, custo de execução, etc.) o escalonamento pode ser ótimo, em caso contrário sub-ótimo. Nesta última categoria, é possível optar pela busca de uma ‘boa’ solução ao invés de procurar no espaço de soluções inteiro (aproximado). Os modelos heurísticos representam a categoria de algoritmos estáticos que assumem um conhecimento *a priori* sobre os processos e as informações do sistema.

Muitas vezes as informações a respeito das tarefas e do sistema não podem ser obtidas antes da execução, levando à utilização de um escalonamento dinâmico.

Dependendo se o escalonador deve residir fisicamente em um único processador ou deve estar fisicamente distribuído entre processadores, classifica-se os escalonadores respectivamente em distribuídos e não-distribuídos (ou centralizados).

Os escalonadores distribuídos são divididos entre aqueles que cooperam entre si e aqueles nos quais cada processador individual toma decisões independentemente das ações tomadas pelos outros. Os escalonadores cooperativos são classificados de forma análoga aos estáticos.

Existem várias características consideradas importantes que não estão presentes na classificação proposta anteriormente. Abaixo estão as principais encontradas na literatura:

- **adaptável e não-adaptável:** a diferença entre elas é o fato de que os adaptáveis mudam dinamicamente seu mecanismo de controle através do uso de parâmetros do sistema, enquanto que os não-adaptáveis não alteram seu algoritmo;
- **alocação única e realocação dinâmica:** na primeira opção os trabalhos são alocados uma única vez a determinado processador, enquanto que na Segunda podem migrar várias vezes antes de serem executados. Claramente, a Segunda opção traz uma sobrecarga maior ao sistema;
- **preemptivo e não-preemptivo:** em um sistema preemptivo uma tarefa pode ser interrompida, uma vez que iniciou sua execução, enquanto que no não-preemptivo isto não é possível;
- **sistema de várias aplicações e de única aplicação:** existem sistemas onde o escalonador está trabalhando com uma aplicação específica, e portanto seu objetivo é minimizar o tempo total para realizar esta aplicação. Por outro lado, podem existir casos onde o escalonamento é para diversas aplicações e deve ser minimizado o tempo médio para resolução de cada aplicação envolvida;
- **balanceamento de carga:** esta política de escalonamento usa a filosofia de que ser justo com o uso dos recursos de *hardware* do sistema é bom para os usuários. A idéia básica é tentar balancear a carga em todos os processadores de maneira que executem a uma mesma taxa. Os nodos agem em conjunto, removendo trabalho de nodos muito carregados para nodos pouco carregados. Esta solução é mais eficiente quando os nodos são homogêneos, pois isso permite que todos os nodos saibam a estrutura do outro;

- **compartilhamento de carga:** esta política de escalonamento tem por objetivo distribuir a carga do sistema de nodos muito carregados para nodos pouco carregados no sistema. A idéia básica é aumentar o desempenho do sistema distribuindo a carga através dos nodos. Não existe a necessidade de que a taxa de execução dos nodos seja semelhante, e sim, de que não existam simultaneamente nodos sobrecarregados e nodos pouco carregados. Esta solução é importante para eficiência tanto em sistemas homogêneos como heterogêneos.

1.2 - Algoritmos de escalonamento

A literatura dos sistemas operacionais fala num sem-número de algoritmos de escalonamento de processos, os mais conhecidos e muito usados na prática são o *round robin*, o das *prioridades*, das *filas multinível*, do *menor job primeiro* e do *escalonamento garantido*.

Antes de analisar algoritmos específicos, devemos pensar naquilo que o escalonador deve buscar. Em primeiro lugar, cabe ao escalonador decidir (baseado numa política) e não fornecer o mecanismo. Os fatores que devem estar presentes em um bom algoritmo de escalonamento, citados anteriormente estão listados abaixo:

- 1) **Justiça:** garantir que todos os processos do sistema terão chances iguais de uso do processador.
- 2) **Eficiência:** manter o processador ocupado 100% do tempo.
- 3) **Tempo de Resposta:** minimizar o tempo de resposta para os usuários interativos.
- 4) **Turnaround:** minimizar o tempo que os usuários batch devem esperar pela saída.
- 5) **Throughput:** maximizar o número de jobs processados na unidade de tempo, usualmente uma hora.

Uma complicação que os escalonadores devem enfrentar é o fato do comportamento de um dos processos ser único e imprevisível. Alguns gastam grande quantidade de tempo fazendo entrada/saída, enquanto outros ocupam o processador por horas a fio, se tivessem a oportunidade.

Quando o escalonador coloca um processo para rodar, ele nunca tem certeza a respeito do tempo que tal processo vai levar executando até ser bloqueado, ou por entrada/saída, ou em um semáforo, ou por qualquer outra razão. Para assegurar-se de que nenhum processo rode por um tempo muito grande é gerado um sinal de interrupção denominado *interrupção de tempo*.

A estratégia de permitir a suspensão temporária de processos que poderiam continuar executando é chamada de *escalonamento preemptivo*, em contraste com a estratégia de *rodar até o fim*, presente nos primeiros sistemas operacionais batch. A utilização de algoritmos não-preemptivos não é adequada para sistemas de propósito geral, onde uma grande quantidade de usuários “competem” os recursos; mas para sistemas dedicados, é razoável que um processo crie um processo-filho e o deixe executar até terminar.

1.3 - Computação Paralela e Sistemas Distribuídos

A *Computação Paralela e Sistemas Distribuídos* tem mostrado certa convergência devido a características e problemas semelhantes (balanceamento de carga, modularidade, tolerância a falhas, etc...). Os *Sistemas Distribuídos* tem sido utilizados para execução de programas paralelos, em substituição das arquiteturas paralelas, pois possuem menor custo e maior flexibilidade.

A utilização desses sistemas computacionais tem-se dimensionado cada vez mais em virtude do crescente avanço tecnológico aliado à necessidade de compartilhamento de informações em diferentes locais.

O desenvolvimento de técnicas e métricas eficientes para a distribuição de tarefas entre os elementos de processamento é ainda um dos grandes desafios existentes em sistemas computacionais distribuídos. Essa atividade de distribuição, conhecida como escalonamento de processos, visa atingir um conjunto de objetivos relacionados com medidas de desempenho, como por exemplo: melhorar o compartilhamento de recursos, otimizar o tempo médio de resposta, promover o balanceamento de cargas, maximizar a utilização dos recursos, entre outros. Esses objetivos são muitas vezes conflitantes e o escalonador, software responsável pela atividade de escalonamento, deve tomar decisões que são influenciadas por diversos fatores: a carga de trabalho do sistema, a presença de aplicações com diferentes características, o hardware da rede de comunicação, o sistema operacional nativo e o hardware dos elementos de processamento.

2) ESCALONAMENTO DE PROCESSOS EM SISTEMAS DISTRIBUÍDOS

Os algoritmos de escalonamento são compostos por políticas e por mecanismos, sendo os últimos responsáveis por manipular os objetivos desses algoritmos. Esses mecanismos são responsáveis pela definição de como o escalonamento será efetuado.

As políticas são responsáveis por definir o que deve ser feito para que ocorra o escalonamento, embora não haja um consenso na determinação dos componentes de um escalonador de processos.

No projeto de um escalonador de processos uma das decisões mais importantes é a escolha entre uma política estática ou dinâmica.

As políticas estáticas não utilizam informações sobre o estado do sistema para escalonar tarefas. Algoritmos estáticos assumem informações sobre os tempos médios de execução de tarefas e necessidades de intercomunicação entre todas as tarefas.

Algoritmos dinâmicos, entretanto, decidem a distribuição de carga no sistema através do estado atual do sistema ou mais recente. Como o estado do sistema sofre alterações dinamicamente, políticas dinâmicas costumam possibilitar melhoras significantes de desempenho.

2.1 - Componentes de um Escalonador

Embora não haja um consenso na determinação dos componentes de um escalonador de processos, os seguintes são bastante difundidos na literatura:

Política de transferência: determina se uma máquina está apta a participar do processo como emissora ou como receptora, conforme sua carga;

Política de localização: responsável por encontrar uma máquina parceira de transferência (emissora ou receptora) adequada para uma máquina, assim que a política tenha decidido que esta máquina é emissora ou receptora;

Política de seleção: escolhe a tarefa a ser transferida (geralmente a iniciada mais recentemente);

Política de informação: decide quando as informações sobre os estados de outras máquina no sistema devem ser coletadas, de onde serão coletadas, e quais informações serão coletadas. Existem três tipos de políticas de informação:

Política orientada à demanda: uma máquina coleta o estado das outras máquina somente quando ela se torna emissora ou receptora;

Política periódica: as informações são coletadas de tempos em tempos;

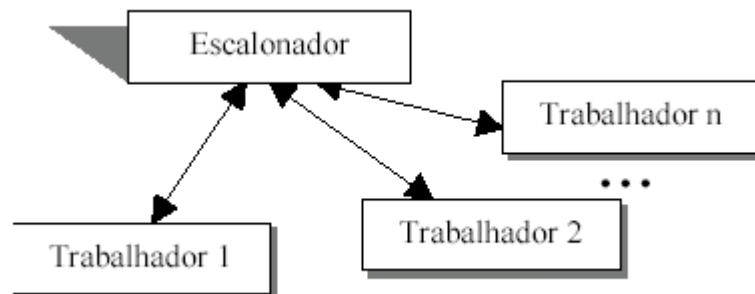
Política orientada à mudança de estado: as informações das máquina são divulgadas conforme mude o grau de seu estado.

Os mecanismos por outro lado podem ser divididos em:

- **mecanismo de métrica da carga:** indica o método utilizado para medir a carga em cada uma das máquinas;
- **mecanismo de comunicação da carga:** define o método através do qual será efetuada a comunicação das informações de carga entre as diversas máquinas;
- **mecanismo de migração:** define o protocolo a ser utilizado quando da ocorrência de migração de processos entre as máquinas.

Dois componentes importantes em um escalonamento dinâmico são a política de transferência e a política de localização. A política de transferência determina quando uma tarefa deve ser processada localmente ou remotamente, enquanto que a política de localização determina o nodo para o qual a tarefa, selecionada para possível execução remota, deve ser enviada. Normalmente, políticas de transferência empregam algum tipo de índice de carga (*threshold*) para determinar quando o nodo está muito carregado ou não.

A maior parte dos sistemas dinâmicos são da categoria fisicamente distribuído, cooperativos, sub-ótimos e heurísticos. Entretanto, o escalonamento dinâmico fisicamente não-distribuído, também denominado centralizado, apresenta importantes vantagens para multicomputadores. A figura 2 apresenta o modelo de escalonamento centralizado.



2.2 - Escalonamento Centralizado

As políticas centralizadas são mais eficientes que as distribuídas para ambientes com um número razoável de processadores em uma rede local (velocidades de comunicação idênticas entre processadores). Políticas centralizadas apresentam problemas de escalabilidade, não

tolerância a falhas, perda de paralelismo, localidade da atividade de escalonamento e informações globais imprecisas.

A escalabilidade diz respeito à capacidade de um sistema paralelo suportar quantidades diferentes de processadores. Normalmente, políticas centralizadas apresentam problemas de “gargalo” quando são submetidas a vários processadores. Escalonadores centralizados não são tolerantes a falhas, uma vez que qualquer falha que ocorra no processador o qual mantém a execução da tarefa escalonadora, causa a interrupção total da aplicação.

Em alguns sistemas com escalonamento centralizado não é possível explorar todo o paralelismo disponível. Isso é devido ao fato de que um único escalonador não consegue detectar e distribuir de forma eficiente todo o paralelismo do problema. O que causa a não detecção do paralelismo é a localidade da atividade de escalonamento (um único processo é responsável).

Como vantagens dos sistemas centralizados são citadas a simplicidade, maior eficiência para um número razoável de processadores, maior eficiência para sistemas com memória compartilhada e menor custo de comunicação.

2.3 - Escalonamento Distribuído

O escalonamento distribuído é aquele em que todos os processadores envolvidos na computação possuem além de trabalhadores, tarefas responsáveis pelo escalonamento. Esta política de escalonamento apresenta um custo de comunicação alto, uma vez que várias mensagens são necessárias para manter a informação de carga global do sistema. Por outro lado, a política distribuída resolve em parte os problemas de escalabilidade e de tolerância a falhas da proposta centralizada. A figura 3 apresenta o modelo de escalonamento distribuído.

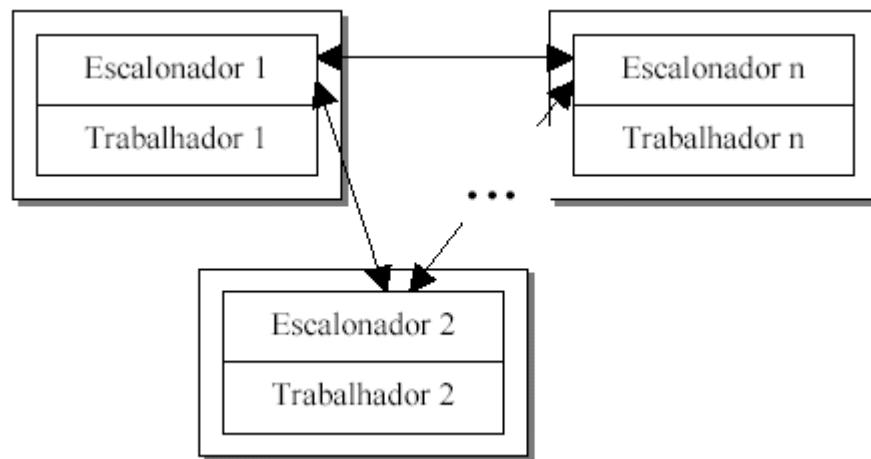


Figura 3 – Escalonamento Totalmente Distribuído

Dentre as principais políticas utilizadas nos sistemas fisicamente distribuídos é possível destacar:

- **bidding**: nesta política de escalonamento os processadores cooperam para que o envio de uma tarefa a um processador beneficie o sistema como um todo. Através de lances (unidades de carga) é escolhido o processador em qual determinada tarefa irá executar.

Uma característica importante desta classe de escalonadores é que todos os nodos geralmente têm autonomia total, uma vez que têm o poder de decidir para onde enviar uma tarefa. Além disso, eles têm autonomia pois não são forçados a aceitar determinado trabalho;

- **probabilísticos:** este esquema é motivada pelo fato de que em muitos problemas o número de permutações entre trabalho disponível e processadores é tão grande que examinar analiticamente o espaço de solução inteiro, demandaria um tempo excessivo de execução. A idéia básica é de escolher aleatoriamente (de acordo com alguma distribuição) processos para serem escalonados. Usando este método repetidamente, um número grande de estratégias de escalonamento podem ser geradas, e então este conjunto é analisado e a melhor escolhida.

Além das políticas apresentadas, várias outras que não se enquadram em categorias específicas são empregadas. Como política de localização para sistemas distribuídos são utilizadas basicamente duas alternativas, a saber: iniciado pelo emissor (*sender-initiated*) e iniciado pelo receptor (*receiver-initiated*).

Nas políticas iniciadas pelo emissor, nodos sobrecarregados procuram transferir trabalho para nodos com pouca carga. Por outro lado, em políticas iniciadas pelo receptor nodos com pouca carga procuram nodos sobrecarregados. A política iniciada pelo emissor comporta-se melhor para sistemas com carga baixa ou moderada. Isso porque, com estas cargas, a probabilidade de encontrar nodos com carga baixa é maior do que a de encontrar nodos com carga alta. Além disso, a política iniciada pelo receptor apresenta grandes problemas de desempenho se a carga não é homogênea no sistema (apenas alguns nodos geram a carga) ou se existe um grande variação de tempos entre o surgimento de novos trabalhos.

Pode-se classificar as políticas de escalonamento distribuído quanto à sua localização física em dois tipos, a saber: totalmente distribuída e hierárquica.

As políticas de escalonamento totalmente distribuído apresentam algoritmos extremamente complexos, uma vez que vários processos idênticos interagem simultaneamente. É importante registrar que os custos de comunicação as tornam proibitivas em arquiteturas fracamente acopladas e com muitos processadores. Muitas vezes, para que sejam obtidos desempenhos satisfatórios, é necessário que apenas alguns nodos (aleatoriamente selecionados) sejam consultados quando da exportação ou importação de tarefas.

A última categoria de política de escalonamento distribuído que será apresentada, a hierárquica, representa um meio termo entre a centralizada e a totalmente distribuída. Ela resolve os problemas de escalabilidade e tolerância a falhas da proposta centralizada sem apresentar os custos de comunicação excessivos da proposta totalmente distribuída. A figura 4 apresenta um modelo de escalonamento hierárquico.

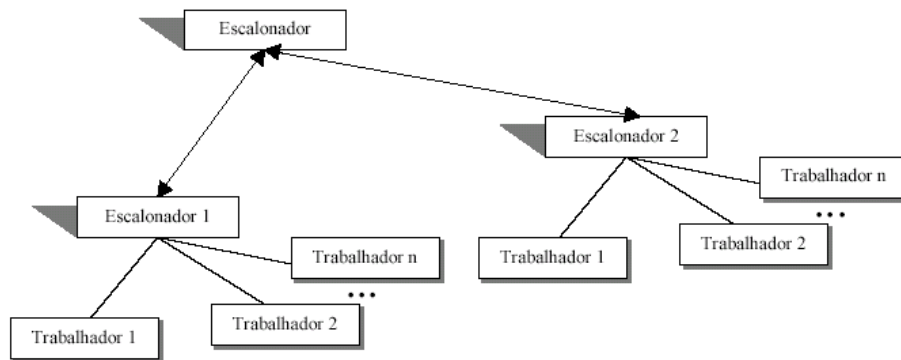


Figura 4 – Escalonamento Hierárquico

A política hierárquica caracteriza-se por possuir escalonadores em alguns processadores do sistema. Estes processadores podem ser dedicados ou não, isto é, podem possuir processos escalonadores e trabalhadores no mesmo processador. Esse tipo de escalonamento possui um conjunto de nodos responsável por manter o estado do sistema, ao invés de um único nodo, como no centralizado. No escalonamento hierárquico, o sistema é dividido logicamente em grupos (*clusters*) e cada grupo de nodos tem um único processo responsável por manter o estado dos nodos do grupo.

Dentre as políticas estudadas a hierárquica apresenta a melhor relação custo/benefício para arquiteturas com memória distribuída. A figura 5 mostra uma tabela que sintetiza as principais características dos três tipos de escalonamento apresentados.

Características:	Centralizado	Totalmente distribuído	Hierárquico
Complexidade dos algoritmos	☹☹☹	☹☹☹	☹☹☹
Escalabilidade	☹☹☹	☺☺☺	☺☺☺
Precisão das informações de carga dos processadores	☺☺☺	☹☹☹	☹☹☹
Quantidade de mensagens	☺☺☺	☹☹☹	☹☹☹
Tolerância a falhas	☹☹☹	☺☺☺	☺☺☺
Desempenho (menos de 100 processadores)	☺☺☺	☹☹☹	☺☺☺
Desempenho (mais de 100 processadores)	☹☹☹	☹☹☹	☺☺☺

Boa → ☺☺☺; Média → ☹☹☹; Ruim → ☹☹☹

3) CONCLUSÕES

Esta monografia apresentou uma introdução ao escalonamento, concentrando-se no escalonamento dinâmico e distribuído. Através deste estudo, foram identificados os principais componentes de um escalonador, bem como feita uma comparação entre as principais políticas de escalonamento existentes.

O escalonamento é uma das tarefas mais importantes em ambientes que exploram o processamento paralelo. Através dele é determinado em quais máquinas e em que ordem os trabalhos serão computados. Várias propostas de escalonamento são encontradas na literatura.

No escalonamento distribuído existe mais de um processador responsável pela tarefa de escalonar. Dentre as várias propostas estudadas é destacado o escalonamento hierárquico. Nesta proposta é obtido um desempenho satisfatório, sem problemas de escalabilidade, excesso de troca de mensagens e tolerância a falhas.

4) REFERÊNCIAS BIBLIOGRÁFICAS

TANENBAUM, A.S. *Modern Operating Systems*. New Jersey, Prentice Hall International, Inc., 1992.

HWANG, K., *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, McGraw-Hill, 1993.

http://lasdpc.icmc.sc.usp.br/pesquisa/kalinka/quali_kalinka.doc