

Segurança do Sistema Operacional e proteção de memória

Vítor Francisco Fonte

vff@di.uminho.pt

João Marco Silva

joaomarco@di.uminho.pt

Universidade do Minho

2019/20

Sistemas Operacionais

Objetivos de segurança importantes:

- separação e controle de acesso aos recursos primeira linha de defesa contra comportamentos indesejados controlador
- fundamental de todos os recursos do sistema comprometê-lo significa perda de controle para os recursos
-
-

Funções importantes:

- controle de acesso a recursos identificação, autenticação e
- gerenciamento de credenciais fluxo de informações e
- sincronização proteção de auditoria e integridade
-

Sistemas Operacionais

Sequência geral de

inicialização:

1. funções primitivas e drivers de dispositivo

2. controladores de processo

3. gerenciamento de arquivos e memória

4. interface do usuário

Sistemas Operacionais

Proteção contra malware:

- deve estar em execução antes de um ataque ocorrer
- mas muitas vezes é um add-on, por isso está sujeito a atraso inicialização

Se um malware conseguir explorar uma vulnerabilidade de sistema operacional

- executar sem ser detectado e como usuário privilegiado (por exemplo, rootkit)

Sistemas Operacionais

Hardware diferente:

- computadores pessoais e mainframes
- dispositivos dedicados automóveis e
- aviônicos smartphones, tablets, aparelhos
- web dispositivos de rede
-

Ajuste adequado para:

- complexidade do dispositivo grau de controle que deve
- exercer quantidade de interação para suportar (humanos,
- dispositivos)

Sistemas Operacionais

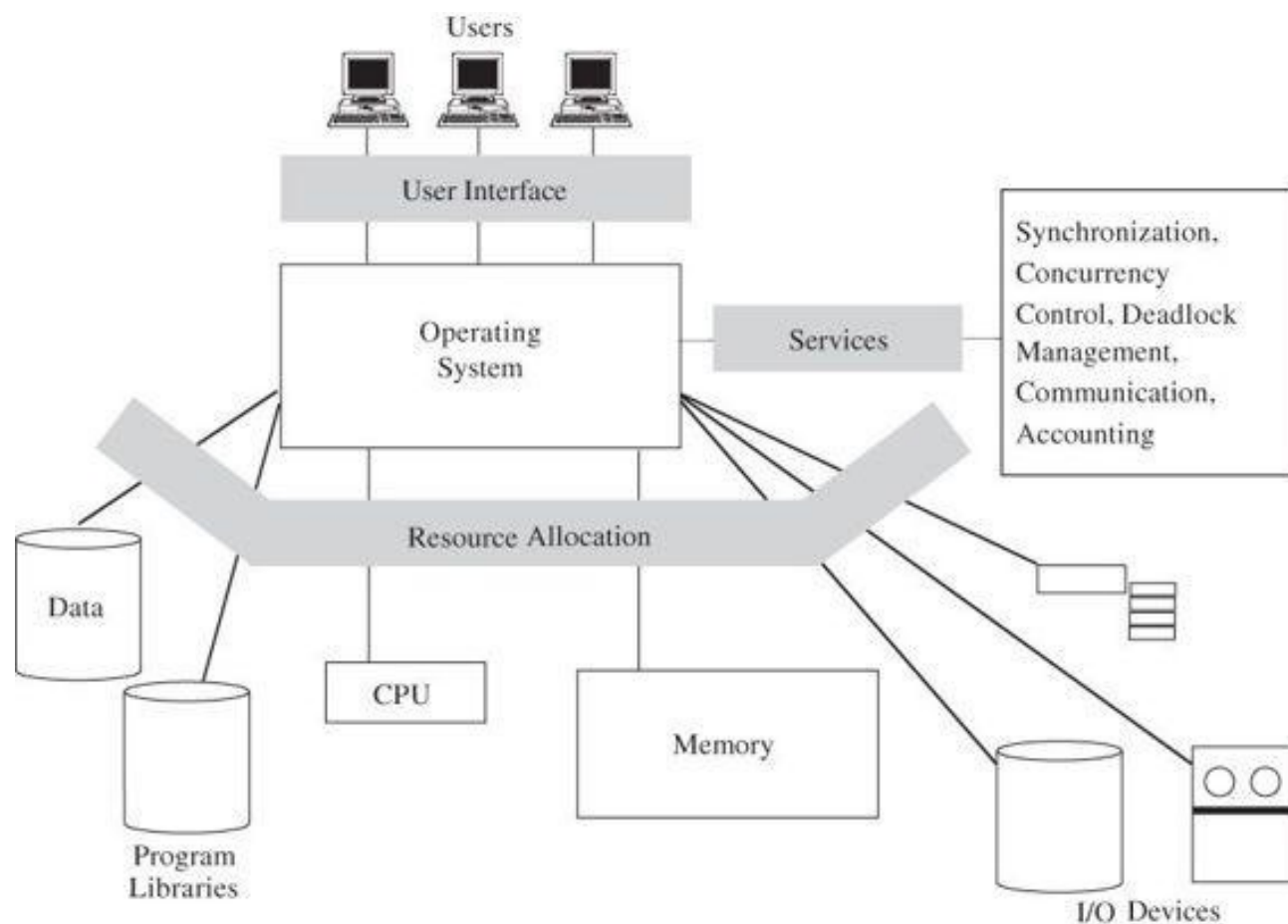
Características:

- único versus multiusuário único versus programa múltiplo
- (agendamento cooperativo ou E/S) única versus multitarefa
- (agendamento preventivo) simples versus multiroscas
-

Do ponto de vista da segurança:

- interessados no controle de recursos do SO quais usuários têm
- permissão de acesso a quais objetos

Sistemas Operacionais



Funções de SO primitivas:

- aka funções do kernel base para a
- aplicação de segurança e outras funções de SO de nível superior

Algumas funções primitivas do SO:

- compartilhamento imposta
- comunicação entre processos e sincronização proteção de dados
- críticos de SO serviço justo garantido
- interface para hardware autenticação
- de usuário proteção de memória
- controle de acesso de arquivos e E/S
- controle de alocação e acesso para objetos gerais
-

Sistema Operacional

Proteção de recursos

Objetos protegidos:

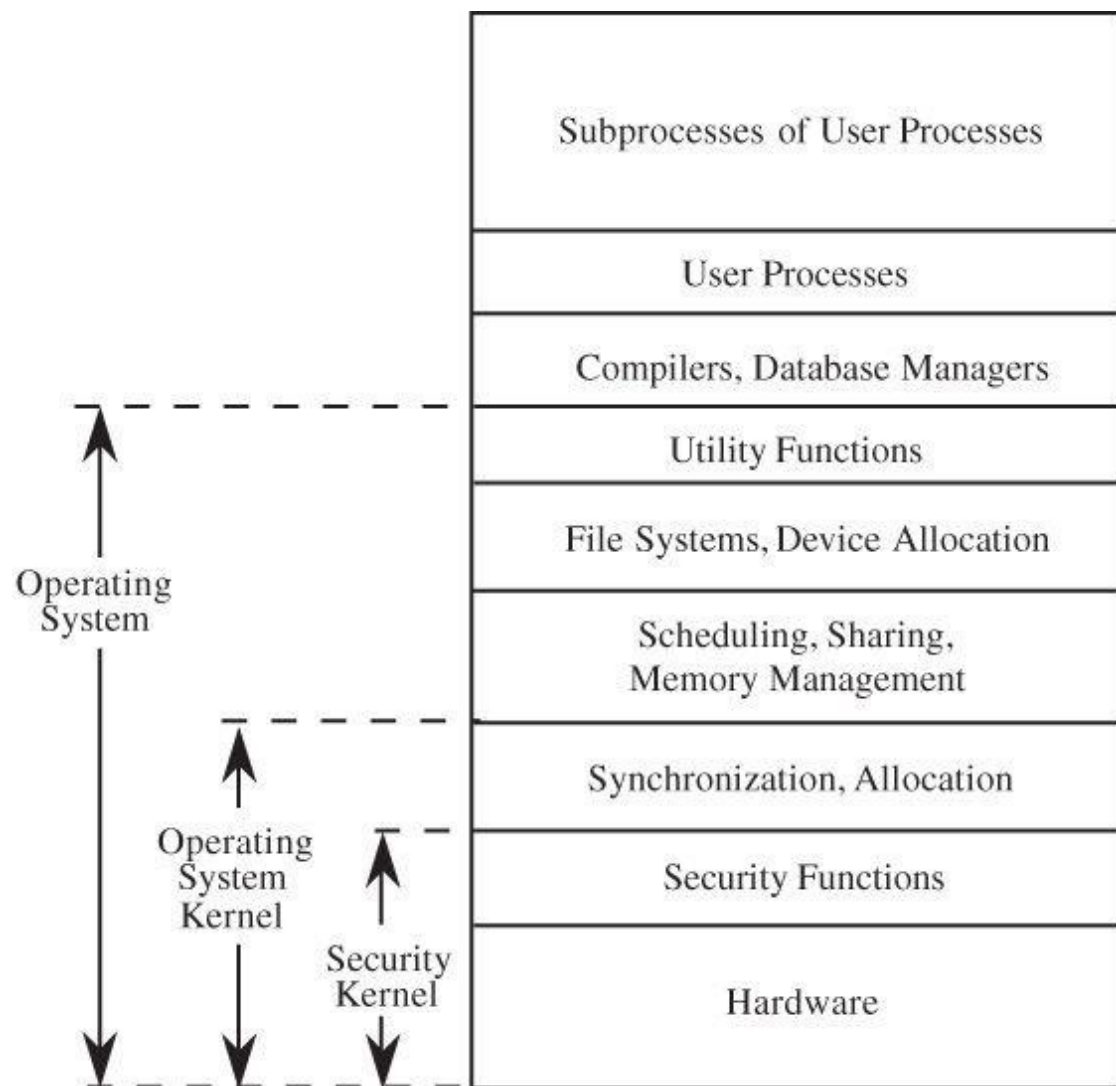
- memória Dispositivos de E/S compartilháveis
- (por exemplo, disco) Dispositivos de E/S
- reutilizáveis serialmente (por exemplo,
- impressora) programas compartilháveis e
- bibliotecas redes dados compartilháveis
-

Mas...

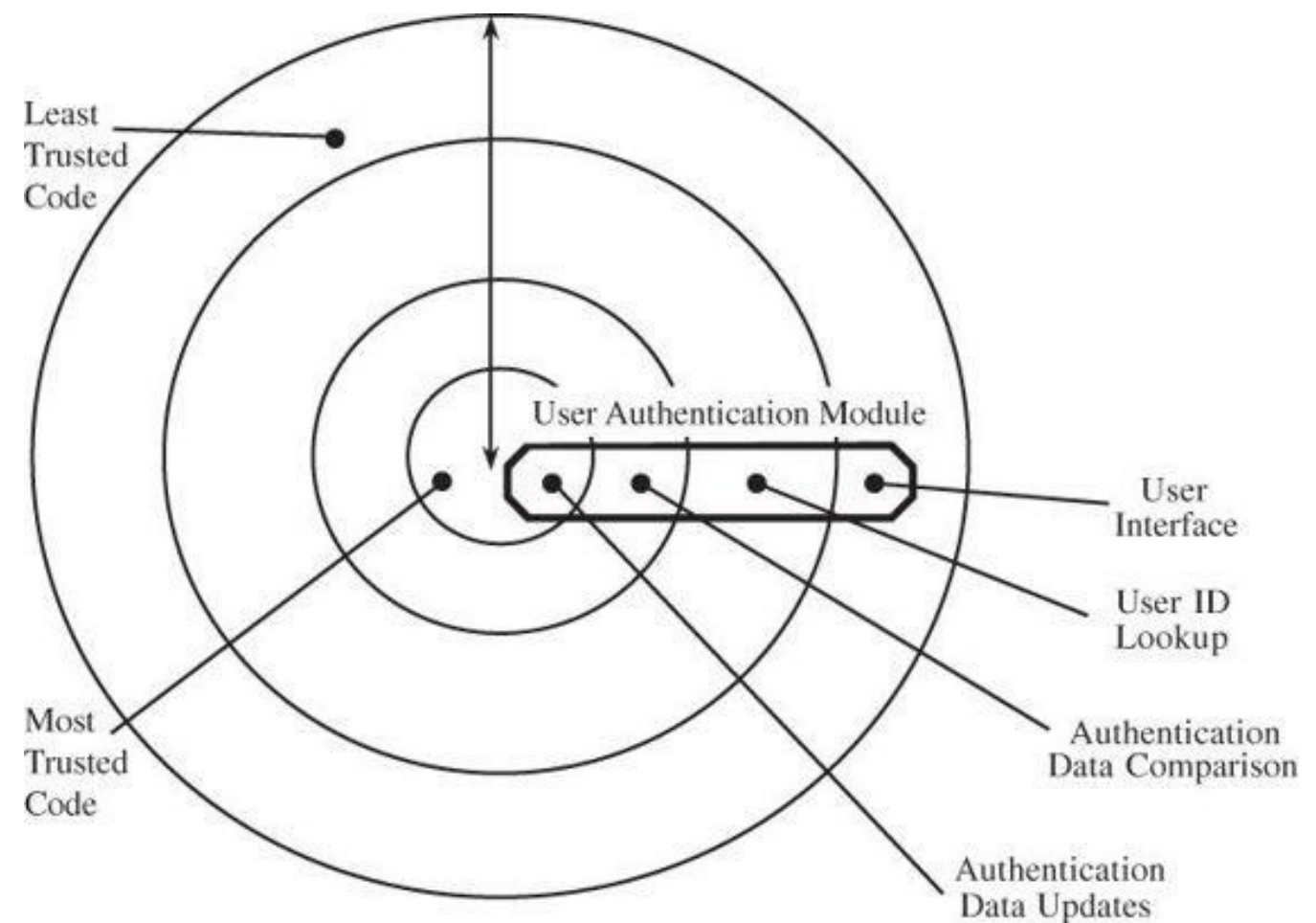
- como proteger esses objetos? como lidar com a necessidade de compartilhar alguns
- desses objetos entre usuários? e em que granularidade recurso?
-

Sistemas Operacionais

Como um sistema em camadas

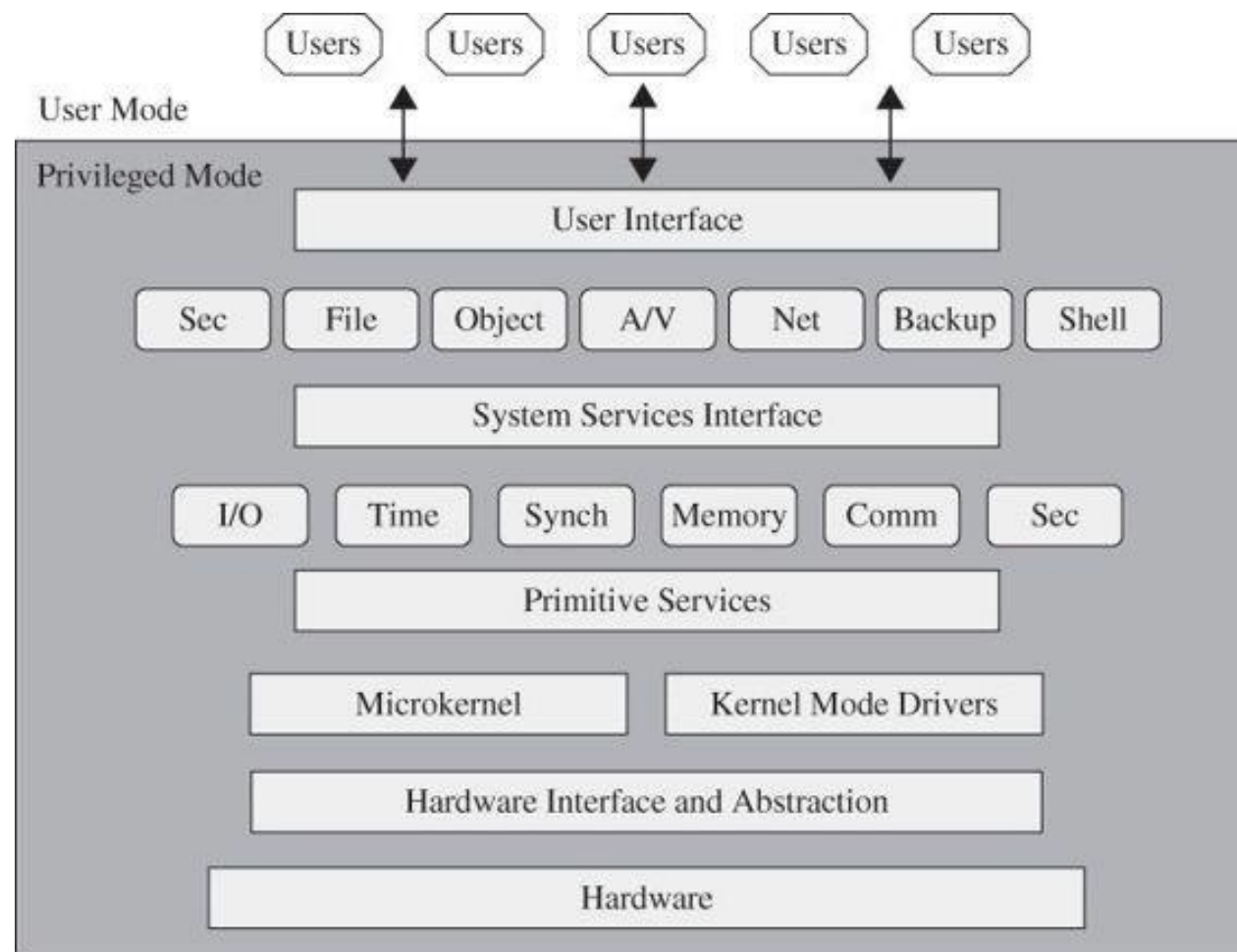


Funções abrangem camadas



Sistemas Operacionais

Como um sistema modular



Segurança e segurança desafios:

- pode vir de Fontes diferentes nem todos
- confiáveis e deve
- integrar com sucesso

Sistemas Operacionais

Design para autoproteção

Um sistema operacional deve ser projetado para auto-- proteção

Contra o compromisso de:

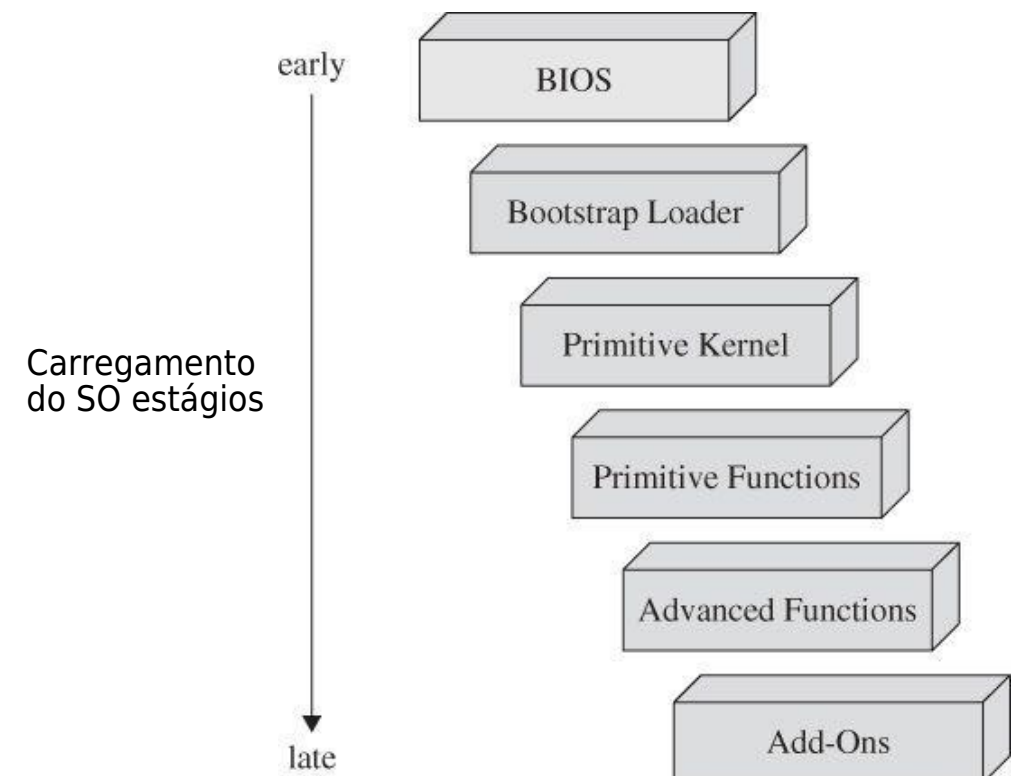
- malicious user programs
- incorporated modules

Desafios:

- tempo, coordenação, hand-off alteração de requisitos e funcionalidade compatibilidade, consistência
-

Conhecimento limitado:

- em quem pode confiar
- para quais recursos



Sistemas Operacionais

Ferramentas para implementar a segurança

Paradigma de controle de acesso:

- Monitor de referência: um sujeito tem permissão para acessar um objeto em um modo particular, e apenas tal autorizado acessado é permitido
- Técnicas de controlo de acesso: lista de controlo de acesso (ACL), lista de privilégios e recursos
- O SO precisa implementar ambas as tabelas subjacentes apoio ao controle de acesso e o mecanismo que verifica se há utilizações aceitáveis

Sistemas Operacionais

Ferramentas para implementar a segurança

Auditoria:

- log: quem, o que, quando, como ferramenta para reagir após
- uma violação de segurança (não a impedindo) quais
- informações foram comprometidas, por quem e quando ajuda
- na prevenção de incidentes futuros nível adequado de
- exploração madeireira útil apenas se pode ser analisado (info
- overload)

Sistemas Operacionais

Ferramentas para implementar a segurança

Virtualização:

- disponibilidade de um conjunto de recursos usando um conjunto
- diferente somente o conjunto de recursos que o usuário tem
- direito a acessar por exemplo, máquina virtual, sandbox,
- honeypot útil para alocação flexível de recursos

Sistemas Operacionais

Ferramentas para implementar a segurança

Hypervisor

- monitor de máquina virtual
implementa uma máquina virtual
- - recebe todas as solicitações de acesso do usuário passa ao longo daqueles que se aplicam
 - a recursos reais que o usuário tem permissão para acessar redireciona outras solicitações
 - para os recursos virtualizados
- separação e sobreposição de recursos vários sistemas
- operacionais e vários hardwares

Sistemas Operacionais

Ferramentas para implementar a segurança

Caixa de areia

- ambiente de execução protegido
- isolados uns dos outros recursos
- frequentemente virtualizados Por
- exemplo, contêineres Docker
 - cada contêiner tem seu próprio software, bibliotecas e configurações comunicar uns com os outros através de canais bem definidos todos os contêineres
 - executados por um único kernel do sistema operacional mais leve do que as máquinas virtuais
 -
 -

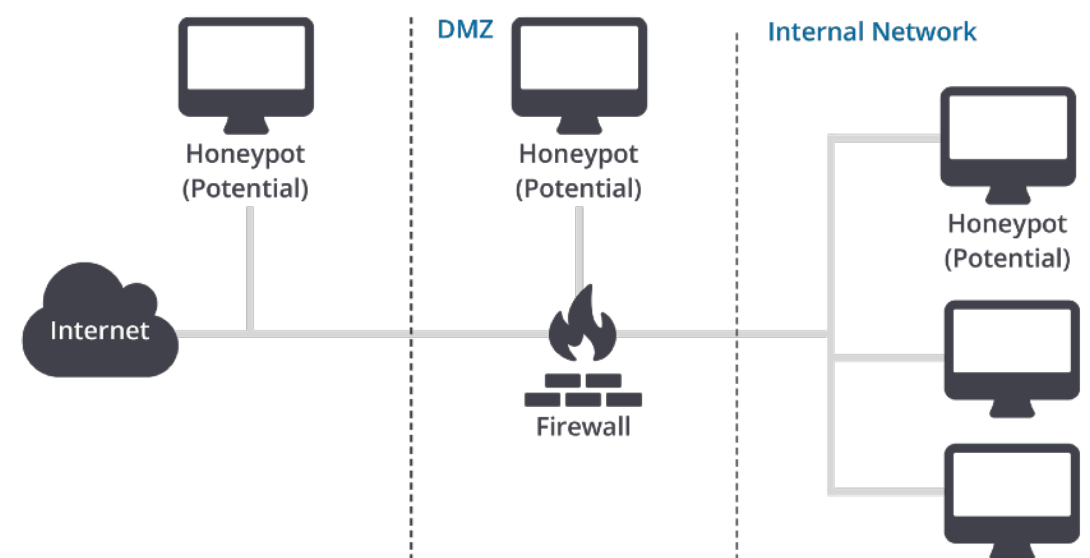
Sistemas Operacionais

Ferramentas para implementar a segurança

Porto de mel

- baseado em vm ou aplicativo Falso, monitorado, rede...
- ambiente acessível destinado a atrair um atacante
 - detectar e/ou desviar um ataque estudar
 - técnicas e objetivos de ataque
- mostra conjunto limitado (seguro) de recursos para o atacante
 - pode sugerir que ele está executando bem conhecido serviços vulneráveis

- tipos: puro, baixo e alto-interação



Exemplo de implantação de potes de mel em um infra-estrutura de rede

Sistemas Operacionais

Ferramentas para implementar a segurança

Potes de mel puros:

- sistemas de produção de pleno direito atacante é monitorado usando um toque de bug instalado
- no link do honeypot para a rede útil, mas furtividade pode ser assegurada por um mecanismo
- mais controlado

Potes de mel de baixa interação:

- simulam apenas os serviços frequentemente solicitados por invasores. relativamente
- poucos recursos, tempo de resposta curto, menos código para implementar, reduzido complexidade da segurança do sistema virtual

Potes de mel de alta interação:

- imitar as atividades dos sistemas de produção que hospedam uma variedade de serviços um invasor pode ter permissão de muitos serviços para desperdiçar seu tempo (por exemplo,
- Honeynet) di ffi cult para detectar, mas caro para manter
-

Sistemas Operacionais

Ferramentas para implementar a segurança

Separação:

- manter os objetos de um usuário separados de outros usuários

Tipos de Separação:

- separação física: processos diferentes usam objetos físicos diferentes de acordo com os requisitos de segurança separação temporal: processos executados em tempos diferentes de acordo com aos requisitos de segurança separação lógica: processos executados como se nenhum outro processo e existem fora de cada
- domínio permitido separação criptográfica: dados de processo e computação são oculto e sem sentido para outros processos
-

Sistemas Operacionais

Ferramentas para implementar a segurança

Observações sobre a separação:

- tipos de separação podem ser usados em conjunto ordem
- crescente de complexidade e ordem decrescente de segurança: física, temporal, lógica muito ínfimo: físico,
- temporal
- eficiência favorece a transferência do ônus da proteção para o sistema operacional

Compartilhamento:

- separação é apenas metade da resposta
- necessidade de compartilhamento de objetos
- entre usuários

Sistemas Operacionais

Ferramentas para implementar a segurança

Diferentes tipos de separação e partilha:

- Não proteja. Adequado quando procedimentos confidenciais são executados em tempos separados. Isole. Processos em execução simultânea não têm conhecimento da presença de um ao outro. Compartilhe tudo ou nada. Proprietário declara que um objeto é público (disponível para todos) ou privado (disponível para o proprietário).
- Compartilhe, mas acesso limitado. OS como proteção entre usuários e objetos, assegurar que apenas ocorre o acesso autorizado. Limitar o uso de um objeto. Não só limita o acesso a objetos, mas também o uso feita dessa aplicação depois de ter sido acedida (por exemplo, ver, mas não copiar, ou copiar, mas não imprimir).
- Observação: ordem crescente de complexidade a implementar e de granularidade
-

Sistemas Operacionais

Ferramentas para implementar a segurança

Proteção de memória por hardware

- separação e partilha compartilhamento de partes da memória
- entre processos e usuários coexistência do sistema operacional
- e dos processos do usuário mecanismos podem ser difíceis de
- implementar mas a maioria pode ser reduzida a hardware a
- partilha pode ser eficiente e resistente a adulterações por
- exemplo: cerca, base/limites, marcado, segmentado, paginado
-

Sistemas Operacionais

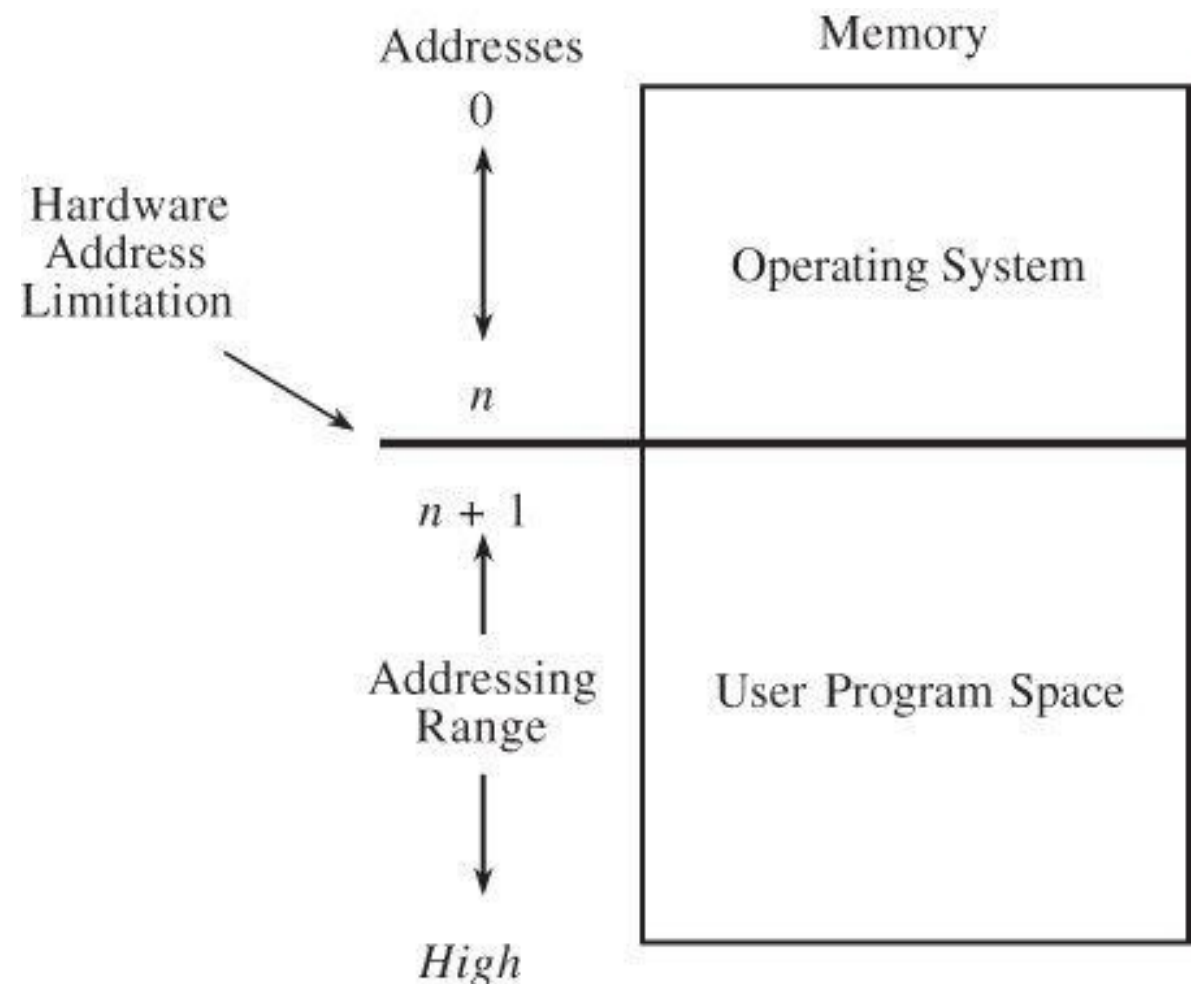
Proteção de memória: cerca

Forma mais simples de proteção de memória:

- introduzido em operação de usuário
- único sistemas um método para limitar
- os usuários a um lado de um limite

Cerca como endereço de memória predefinido:

- uma quantidade predefinida de espaço reservado para SO muito
- restritivo

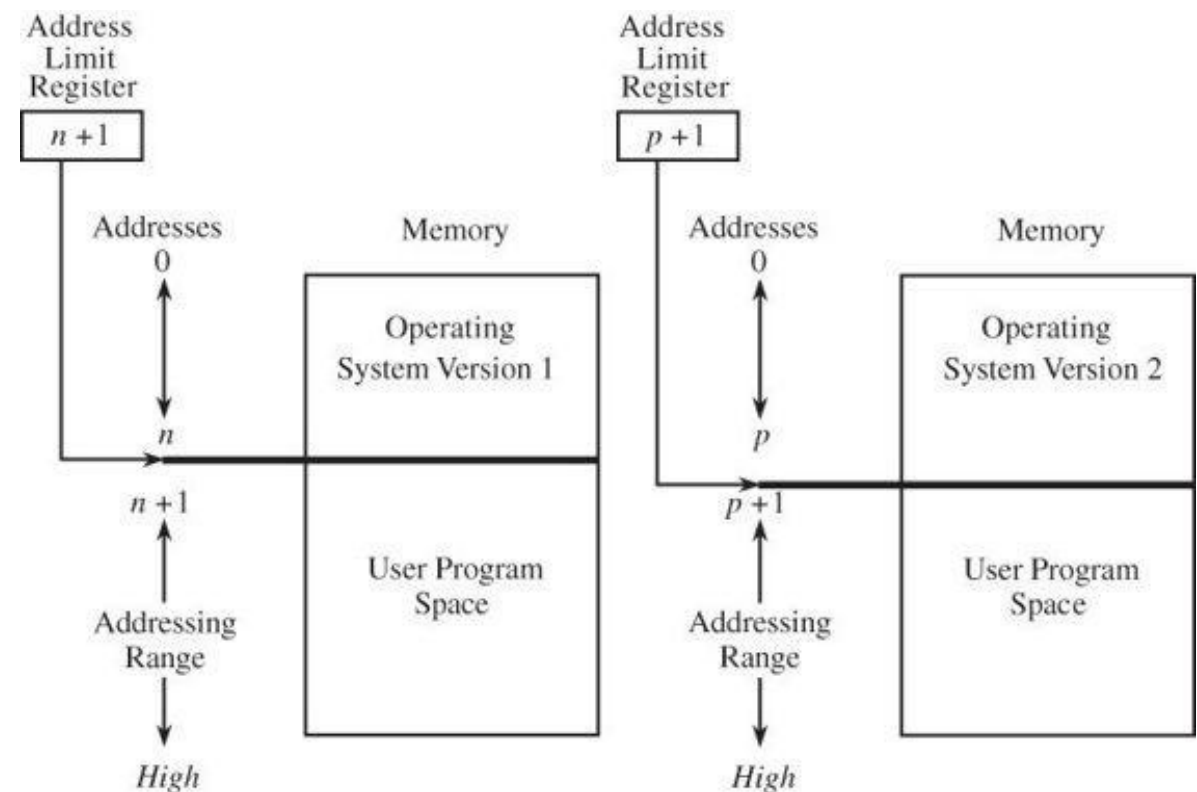


Sistemas Operacionais

Proteção de memória: cerca

Conjunto de vedação no registro de hardware:

- quantidade de espaço para o sistema operacional pode ser
- alterado acesso à memória endereços é automaticamente em comparação com o registro de cerca se maior fino o acesso foi concedido, se menor foi negado



Sistemas Operacionais

Proteção de memória: cerca

Protege apenas em uma direção:

- o sistema operacional pode ser protegido contra os usuários
- programas de usuário não podem ser protegidos uns dos outros
- programas de usuário não podem proteger áreas do programa:
- - por exemplo, definir áreas como não-graváveis ou não-executáveis

Sistemas Operacionais

Proteção de memória: Registros Base/Limites

Registros de base/limites:

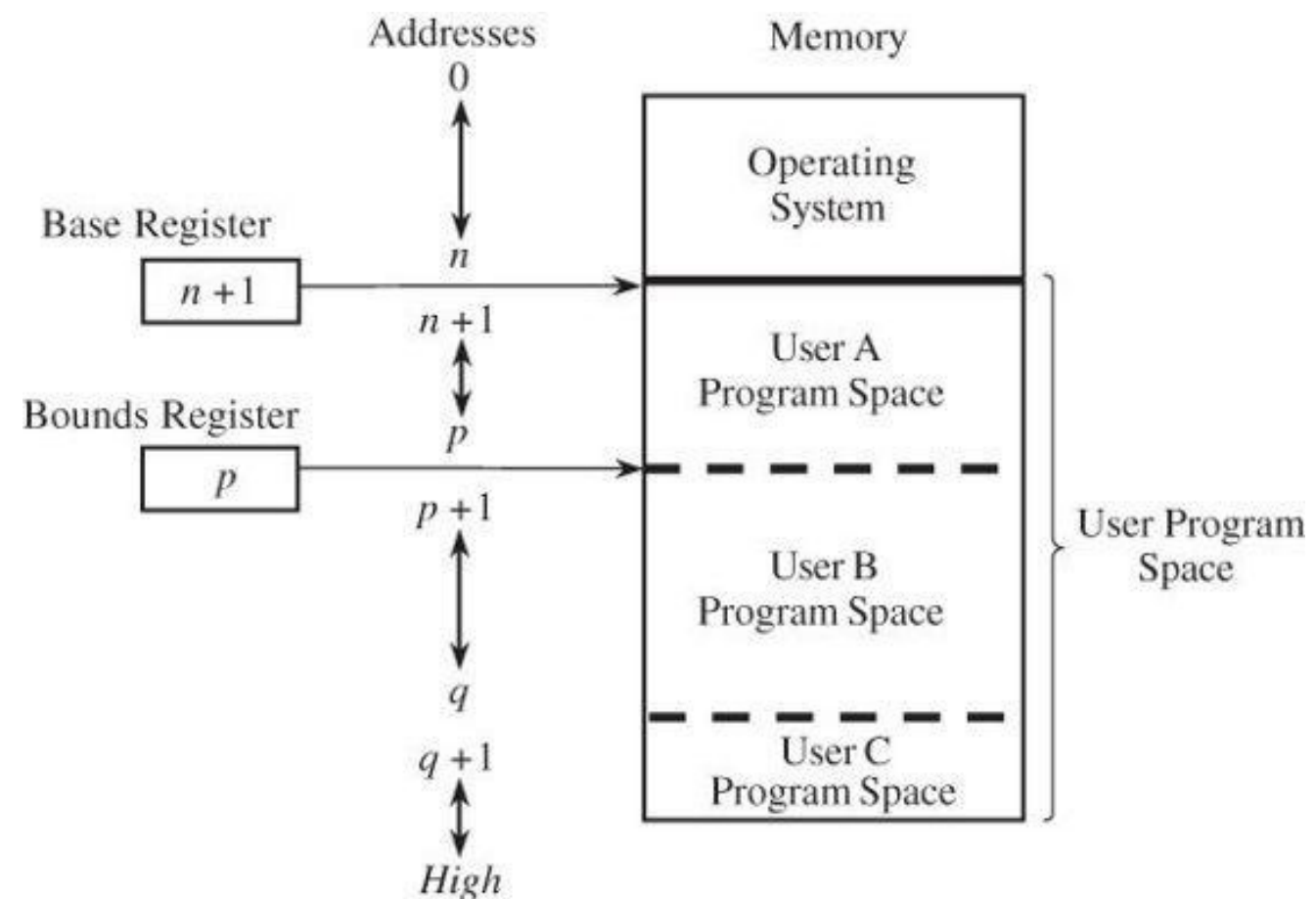
- rodeia um espaço de endereço do programa específico para cada programa de usuário
- atualizado sobre a alternância de contexto
-

Endereços do programa do usuário:

- sempre adicionado a um registro base sempre verificado contra um registro de limites
-

Proteção e realocação:

- importante para sistemas multiprogramação



Sistemas Operacionais

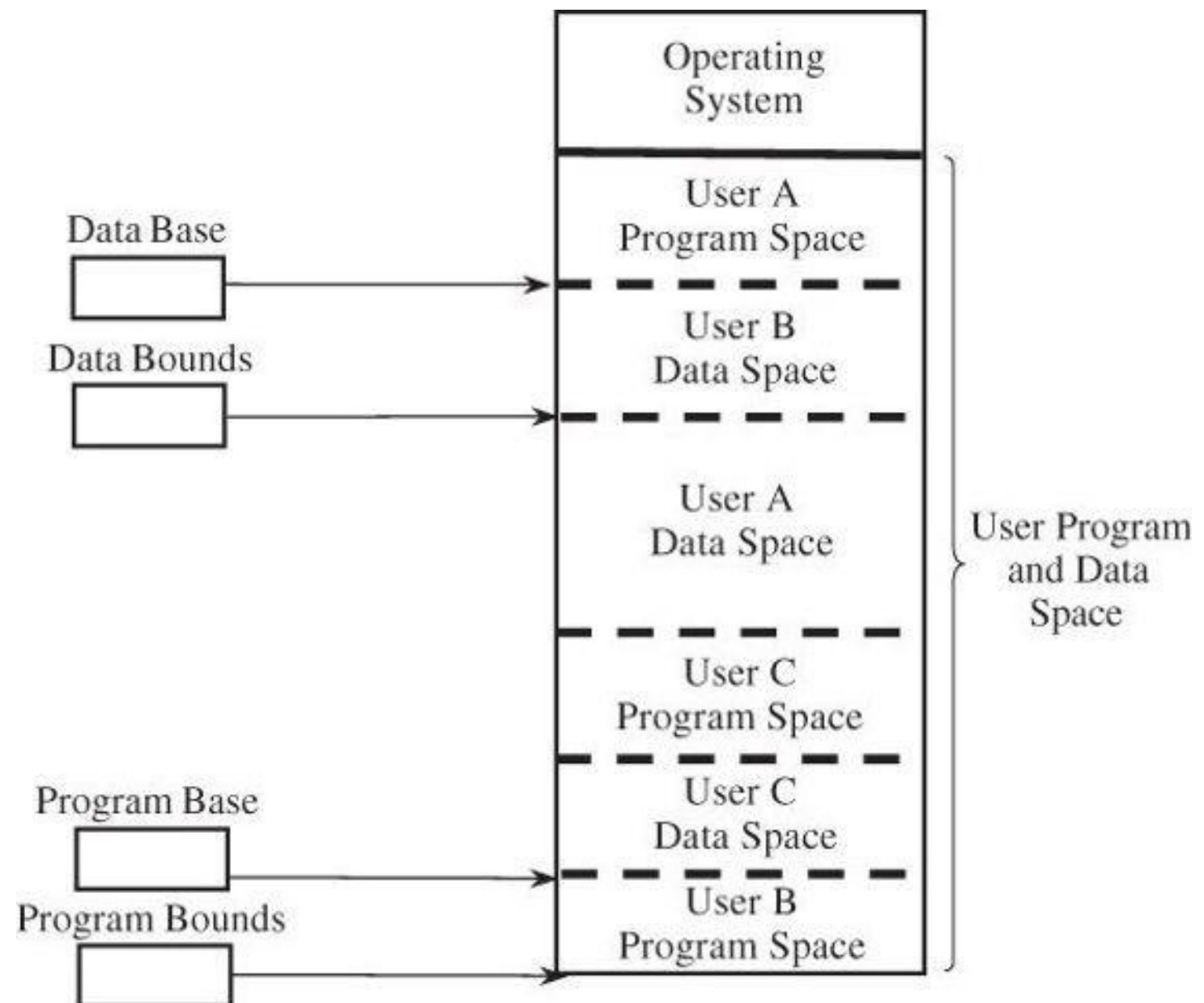
Proteção de memória: registros base/vinculados

Registros base/limites podem ser usado para proteger específicos áreas de memória do usuário programas:

- por exemplo, código e dados

Pode ser estendido para suportam mais de dois áreas

- mas limite prático é de dois registrar apenas pares

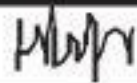
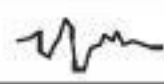
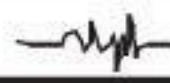
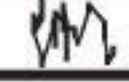
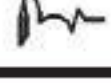
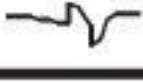


Sistemas Operacionais

Proteção de memória: Arquitetura marcada

Arquitetura etiquetada:

- cada palavra tem bits extras identificando os direitos de acesso a essa palavra (ou intervalo de palavras) bits extras são definidos apenas por instruções privilegiadas
- (SO) estes bits são verificando em cada acesso à palavra por razões
- práticas o número de bits era sempre pequeno não popular devido
- à compatibilidade desafios

| Tag | Memory Word |
|-----|---|
| R | 0001 |
| RW | 0137 |
| R | 0099 |
| X |  |
| X |  |
| X |  |
| X |  |
| X |  |
| X |  |
| R | 4091 |
| RW | 0002 |

Code: R = Read-only RW = Read/Write
X = Execute-only

Sistemas Operacionais

Proteção de memória: memória virtual

Memória virtual:

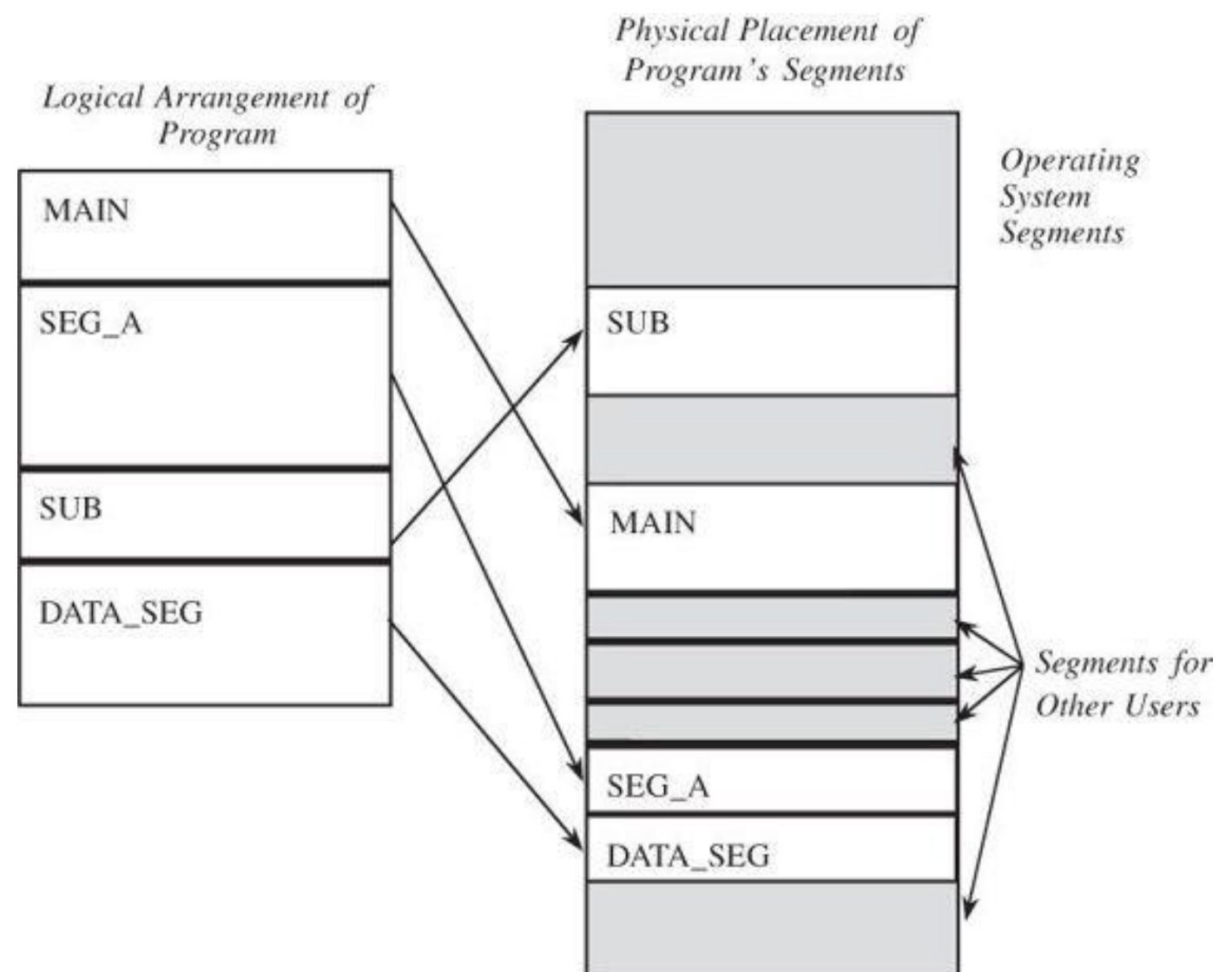
- suporte a hardware geralmente (por exemplo, MMU) usado na maioria dos sistemas operacionais de uso geral vantagens no
- endereçamento e proteção por exemplo, segmentação, paginação, paginação com segmentação
-
-

Sistemas Operacionais

Proteção de memória: segmentação

Segmentação:

- programa dividido em lógica unidades (tamanhos diferentes)
- endereçamento: nome de conjunto de segmentos mais o ff semelhante a um “não consolidado” número de pares de base/limites processo: o
- sistema operacional mantém um tabela de segmento acessível nomes
- map: nomes de segmentos e endereços reais (e tamanho)

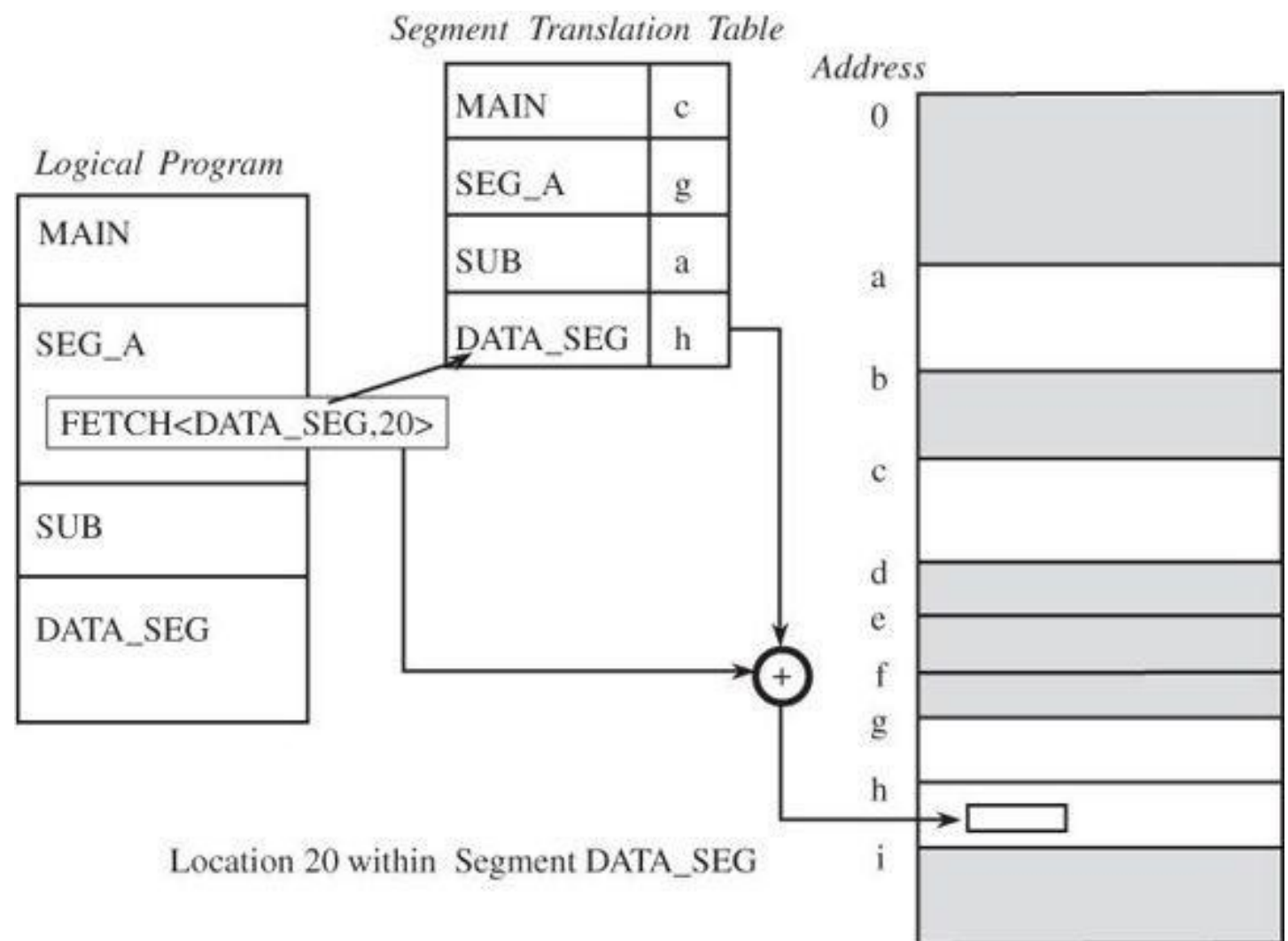


Sistemas Operacionais

Proteção de memória: segmentação

Segmentação:

- vantagens: deslocalização, compartilhamento, troca, proteção (em cada acesso, MMU), tradução rápida de endereço, não fragmentação interna
- desvantagens: inconveniência (por exemplo, endereço deve identificar um segmento), fragmentação externa



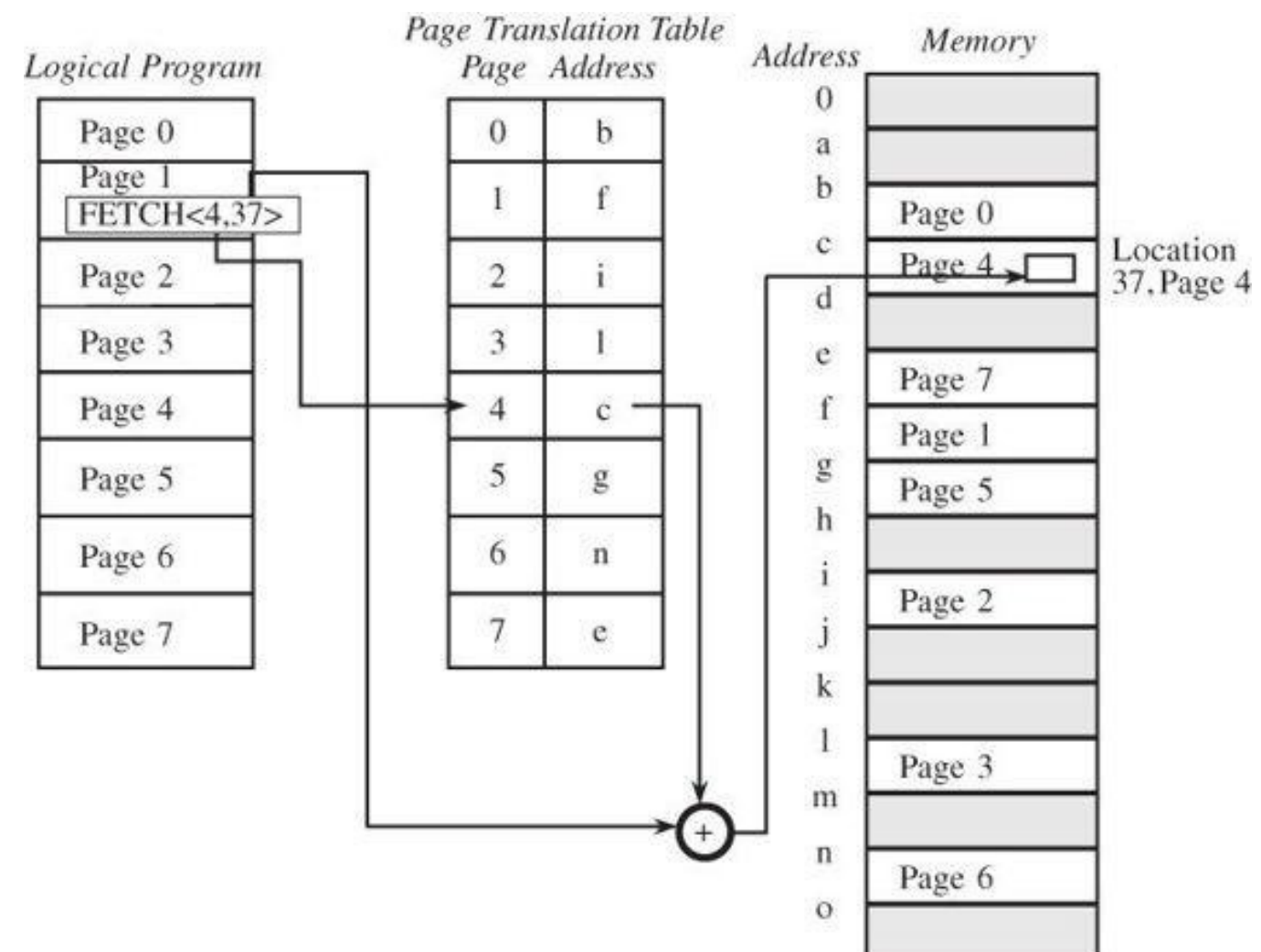
Sistemas Operacionais

Proteção de memória: paginação

Paginação:

- programa dividido em igual - unidades de tamanho (quadro, por exemplo, 4 KB) endereçamento:
- quadro de página (número) mais o conjunto ff processo: o sistema operacional mantém uma tabela de quadros de página acessíveis mapa: quadros de página e real endereços muito baixo interno e nenhum fragmentação externa vantagem:
- nenhum interno ou fragmentação externa
-
-

- desvantagem: mais complexo, mais lento do que a segmentação, nenhuma unidade lógica



Sistemas Operacionais

Proteção de memória: Segmentação paginada

Segmentação paginada:

- processo: conjunto de lógica segmentos
- segmento: conjunto de fixos tamanho de páginas
- vantagens: flexível tamanhos de página, simplificado alocação de memória, nível adicional de proteção

