

Aluno: João Victor da Silva Prado

Sistemas Operacionais

Atividade 7

1) R: Como sabemos, durante sua execução, um programa precisa usar a memória. Cada processo des-
ses é visto pelo sistema operacional como uma área
onde só ele e o núcleo vão poder acessar. Para
um processo haverá o endereço inicial(0) e o end-
reço máximo(n), com divisões de faixas de end-
reço que são mantidas pelo núcleo: TEXT, DATA,
HEAP, STACK, BSS.

- TEXT: Onde o código executável (programa princi-
pal) do processo é armazenado, ou seja, onde estão
as bibliotecas e funções, ou melhor, onde asinstru-
ções ficam.

- DATA: Onde variáveis estáticas e buffers internos
estão.

- BSS: O segmento de dados não inicializados. Os
dados nesse segmento são inicializados pelo kernel
para "0" antes do programa começar a executar os
dados não inicializados no final do segmento de da-
dos e contêm todas as variáveis globais e variáveis
estáticas que são inicializadas com zero ou sem
inicialização explícita no código fonte.

- Em seguida há uma estrutura de tamanho flexível
que varia de acordo com cada processo. Ela vai da
faixa chamada HEAP (onde são armazenadas as variáveis
dinâmicas) à faixa chamada STACK (endereços de re-
torno de chamadas).

2) R: Levando em consideração o uso de hardware dedicado, mmu, há a definição ou resolução de endereço pelo processador durante a execução. Isso acaba criando endereços de memória chamados de endereços lógicos, que não são necessariamente os mesmos que os associados à memória principal, também chamados de endereços físicos.

O uso de endereços lógicos pelo mmu permite que, quando existir múltiplos processos ativos em um sistema, eles sejam separados e evite a sobreposição da informação. Isso ocorre porque embora eles sejam alocados na mesma faixa de endereço, na memória principal serão alocados em lugares diferentes. Para facilitar a compreensão das diferenças:

	End. lógico (endereço virtual) (gerado pela CPU)	End. físico (local numa unidade de memória)
espaço de endereçamento	Conjunto de todos os endereços lógicos gerados pela CPU em referência a um programa.	Conjunto de todos os endereços físicos mapeados para os endereços lógicos correspondentes.
visibilidade	O usuário pode visualizar o endereço lógico de um programa.	O usuário nunca pode ver o endereço físico do programa.
acesso	O usuário usa o endereço lógico para acessar o físico.	O usuário não pode acessar diretamente o end. físico.
geração	O endereço lógico é gerado pela CPU.	O endereço físico é calculado pela mmu.

3) 0: 20 → 64

1: 75 → 275

2: 90 → 90

3: 1350 → 3350

4: 510 → 1710

4)

a) 0x0877

0000 100001110111

Pg 0 offset: 2167

b) 0x4BAC

0100 101110101100

Pg 4 offset: 2988

c) 0x75FD

0111 0101111101

Pg 7 offset: 1933

d) 0x0AD6

0000 101011010110

Pg 0 offset: 2774

5) R: A falta de página (ou Page fault) se trata de uma exceção feita pelo hardware quando um programa acessa uma página mapeada na memória virtual que não foi carregada na memória física. Suas causas podem variar desde o caso em que a página correspondente ao endereço solicitado não está carregada na memória, ao caso em que apesar de estar carregada o seu estado corrente não foi atualizado no hardware. O tratamento é realizado através da interrupção do processo em si ou realizando uma alocação da memória física.

6) R:

a) Incorreta, na verdade é o contrário. O processador gera endereços lógicos e esses endereços são convertidos pela MMU em endereços físicos.

b) Correto.

c) Correto

d) Incorreta, pois o armazenamento é apenas das variáveis estáticas inicializadas pelo código fonte

e) correto

f) Incorreta, pois não há threads armazenadas na área de memória STACK pela dinamicidade e atenção requerida. A área de memória STACK aloca dados ou ponteiros quando uma função é chamada e desalocado quando essa função termina.



D S T Q Q S S

7) * Páginas de 3 níveis; Tempo de acesso à memória RAM = 100 ns; TLB 64 entradas; 98% de taxa de acerto; 10 ns de custo de acerto e 50 ns de erro.

* O tempo de acesso à memória pode ser determinado pela média ponderada entre o tempo de acesso com acerto de cache e o tempo de acesso no caso de erro. Nesse tempo médio serão considerados os custos de acerto e erro, de consulta às tabelas e tempo de acesso à RAM.

$$\text{Tempo médio} = (0,98 \cdot 10 \text{ ns}) + (0,02 \cdot 50 \text{ ns}) + (0,02 \cdot 3 \cdot 100 \text{ ns}) + 100 \text{ ns}$$
$$\text{Tempo médio} = \underline{116,8 \text{ ns}}$$