

SO

TESTE 1 – PERGUNTA 19/23/14/28/3/31/17/33/27/8/11/16 depois da 34 ao fim

TESTE 2 – PERGUNTA 1 Á 33

1.A partição de um disco permite ao utilizador:

- a. Manter diferentes sistemas de ficheiros no mesmo dispositivo.
- b. Partilhar a informação no dispositivo com outros utilizadores.
- c. Diminuir a latência na transferência de informação para a memória.

2.A System-Wide Open File Table (SWOFT) é uma tabela mantida pelo kernel em memória contendo informação relativa a todos os ficheiros:

- a. Guardados atualmente nos discos.
- b. Que alguma vez estiveram nos discos.
- c. Abertos por processos em execução.

3.Como explica que o seguinte programa dê valores distintos para a variável “val” que está localizada no mesmo endereço?

```
/* includes & defines */

int main (int argc, char* argv[]){
    pid_t pid;
    int val;

    pid = fork();

    if (pid == 0){
        val = 0;
        printf(" child: val = %d, at addr = %p\n", val, &val);
        return EXIT_SUCCESS;
    } else{
        val = 1;
        wait(NULL);
        printf("parent: val = %d, at addr = %p\n", val, &val);
        return EXIT_SUCCESS;
    }
}

$ gcc -Wall val.c -o val
$ ./val
child: val = 0, at addr = 0x7ffeef275a38
parent: val = 1, at addr = 0x7ffeef275a38
```

- a. “val” é alterada na memória física no mesmo endereço em instantes diferentes.
- b. Há um erro no programa, depois do “fork” os endereços têm de ser diferentes.
- c. Os espaços de endereçamento dos processos são iguais e o endereço é virtual.

4.Os algoritmos FCFS (First Come First Served), SSFT (Shortest Seek Time First), SCAN e outros são usados pelo sistema operativo para organizar as operações de leitura/escrita em discos HDD tendo em vista:

- a. Minimizar a rotação dos pratos.
- b. Minimizar o movimento das cabeças.
- c. Minimizar o tempo de transferência.

5.As instruções do CPU operam directamente apenas com informação na memória física, e nunca com informação guardada em discos, porque:

- a. A latência de acesso é muito superior nos discos do que na memória.
- b. Necessitariam de endereços virtuais enormes para aceder à informação.
- c. Porque o CPU e os discos não estão ligados directamente na motherboard.

6.Relativamente a um disco rígido (HDD), um disco de estado sólido (SSD):

- a. Tem um preço mais baixo por GB de capacidade.
- b. Tem um tempo médio de utilização mais longo.
- c. Tem uma maior largura de banda para a memória.

7.Sistema de indexação do UNIX (inode) está optimizado para o cenário de muitos ficheiros de pequeno tamanho porque:

- a. Ficheiros pequenos são indexados directamente pelo inode e os inodes têm identificadores de 32 e 64 bits.
- b. Suporta apenas indexação directa de ficheiros pelo inode e estes têm identificadores no máximo de 8 bits.
- c. Suporta níveis 0, 1, 2 e 3 de indexação para ficheiros e os inodes têm identificadores de 32 e 64 bits.

8.Se imprime o valor de um apontador obtendo, por exemplo, 0x7ffeef275a38, quantos bits tem a arquitectura em que o programa foi executado?

- a. 12 bits.
- b. 64 bits.
- c. 32 bits.

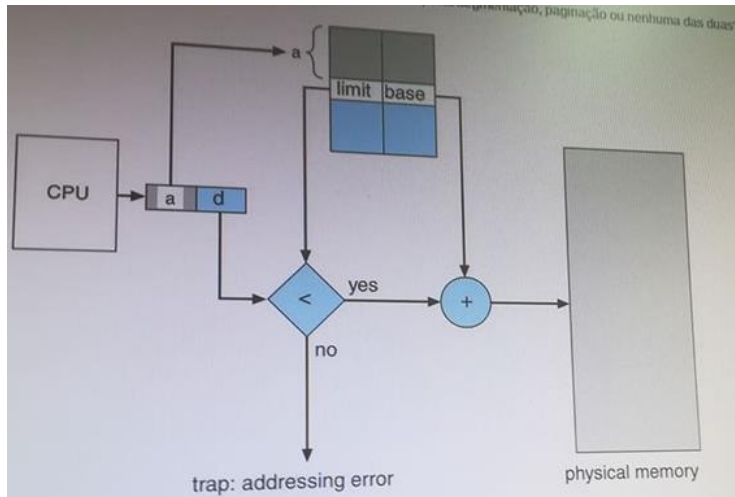
9.A função de um controlador de um disco é:

- a. Receber dados de periféricos de I/O, guardá-los, e sinalizar o CPU através de interrupts.
- b. Garantir que um disco não lê nem escreve mais do que o indicado pelos comandos CPU.
- c. Recebe do CPU, via motherboard, comandos de leitura/escrita de dados e executá-los.

10.Um volume é:

- a. Uma partição de um disco usada para transferir páginas de/para memória.
- b. O conjunto de todos os sistemas de ficheiros instalados num disco.
- c. Uma partição de um disco onde foi instalado um sistema de ficheiros.

11.A imagem que se segue representa uma MMU. O circuito suporta segmentação, paginação ou nenhuma das duas? Porquê?



- Paginação, porque a tabela de tradução tem o valor “limit” para cada entrada, indicando que os limites das páginas estão a ser verificados.
- Nenhuma, porque a existência da tabela com valores “limit” e “base” por entrada indica que esta guarda a posição de processos completos na memória.
- Segmentação, porque a tabela de tradução tem o valor “limit” para cada entrada, indicando que os blocos de memória podem ter tamanhos diferentes.

12.As estruturas de dados Per-Process Open File Table (PPOFT), System-Wide Open File Table (SWOFT) e File Control Block (FCB) são mantidas na memória pelo kernel do sistema operativo para:

- Diminuir a latência das operações sobre o sistema de ficheiros.
- Diminuir o custo energético de aceder sempre a informação nos discos.
- Manter ficheiros para lá do limite de capacidade dos discos.

13.A estrutura em C seguinte (exemplo do MACOS X) contém informação normalmente guardada:

```

struct stat {
    dev_t    st_dev;    /* device inode resides on */
    ino_t    st_ino;    /* inode's number */
    mode_t   st_mode;   /* inode protection mode */
    nlink_t  st_nlink;  /* number of hard links to the file */
    uid_t    st_uid;    /* user-id of owner */
    gid_t    st_gid;    /* group-id of owner */
    dev_t    st_rdev;   /* device type, for special file inode */
    struct timespec st_atimespec; /* time of last access */
    struct timespec st_mtimespec; /* time of last data modification */
    struct timespec st_ctimespec; /* time of last file status change */
    off_t    st_size;   /* file size, in bytes */
    quad_t   st_blocks; /* blocks allocated for file */
    u_long    st_blksize; /* optimal file sys i/o ops blocksize */
    u_long    st_flags; /* user defined flags for file */
    u_long    st_gen;   /* file generation number */
};
  
```

- Num File Control Block (inode Unix).
- Numa entrada da Per-Process Open File Table.
- Numa entrada da System-Wide Open File Table.

14. Uma vantagem/desvantagem, respetivamente, de usar páginas mais pequenas num sistema operativo é:

- a. Diminuir a fragmentação interna / aumentar o “swapping” de páginas.
- b. Diminuir a fragmentação externa / diminuir a localidade de informação.
- c. Aumentar a localidade de informação / aumentar a fragmentação externa.

15. File Allocation Table (FAT) usa lista de blocos para guardar o conteúdo de ficheiros mas é mais eficiente porque a tabela FAT contém:

- a. Os blocos mais usados de cada ficheiro e é mantida em memória pelo kernel.
- b. Todas as ligações entre blocos ficheiros e é mantida em memória pelo kernel.
- c. Todos os primeiros blocos de ficheiros e é mantida em memória pelo kernel.

16. Na técnica da segmentação:

- a. Os endereços traduzidos são sempre válidos no programa.
- b. A MMU é simples pois os segmentos são do mesmo tamanho.
- c. Não ocorre a fragmentação externa da memória física.

17. A diferença fundamental entre arquitecturas de 32 e 64 bits consiste no facto de as segundas permitem:

- a. A execução de processos com espaços de endereçamento maiores.
- b. Transferir processos completos para a memória física.
- c. Manter informação sobre mais processos na memória física.

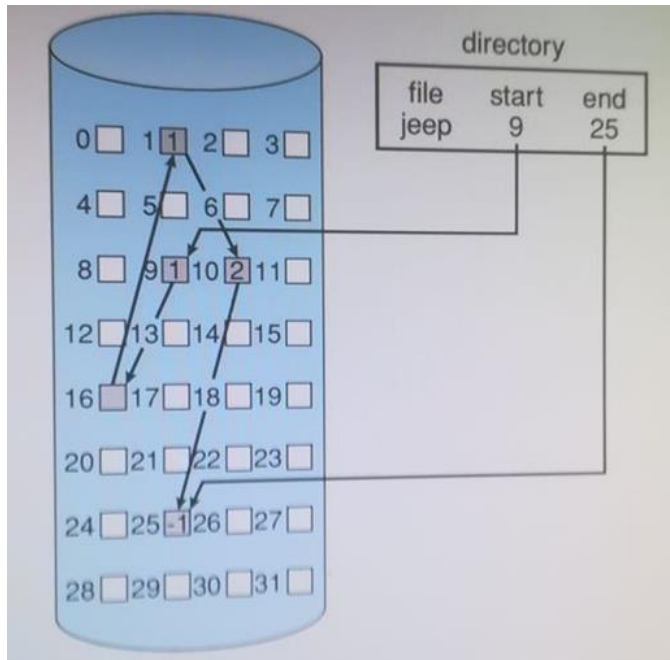
18. Num disco rígido (HD) a latência no acesso aos dados:

- a. Não é uniforme devido ao movimento dos pratos e das cabeças de leitura/escrita.
- b. Não é uniforme devido à capacidade variável do buffer de leitura/escrita do controlador.
- c. É uniforme, i.e., demora o mesmo tempo obter informação em qualquer posição do disco.

19. Uma vantagem significativa da técnica da paginação é:

- a. Todos os endereços traduzidos são válidos no programa.
- b. As páginas têm tamanhos variável e estritamente necessário.
- c. Não provoca a fragmentação externa da memória física.

20. A utilização do mecanismo de gestão do espaço em disco para ficheiros ilustrado na figura seguinte:

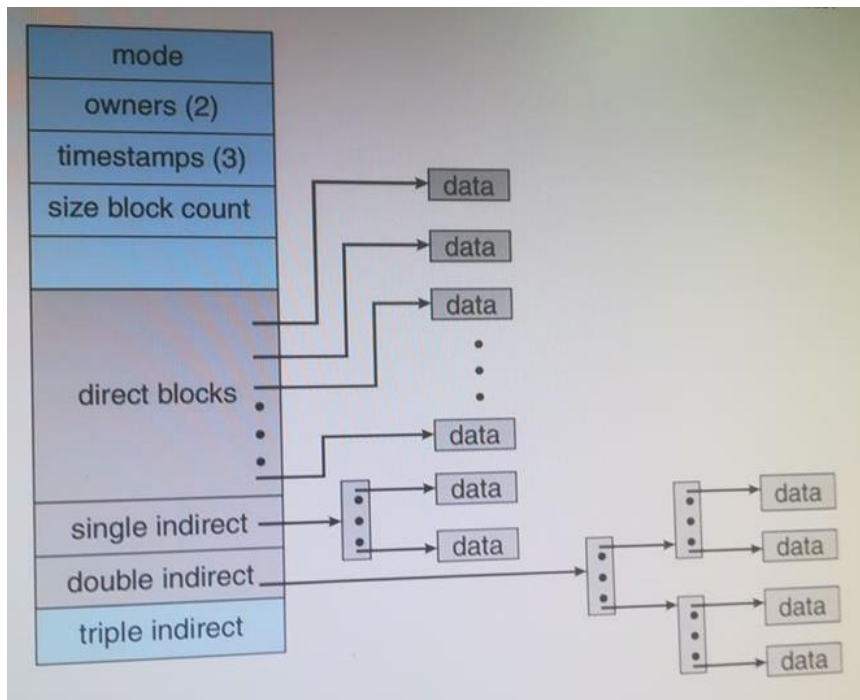


- a. **Incorre na latência mais elevada para acessos aleatórios.**
- b. Dificulta a criação de ficheiros grandes e expansíveis.
- c. Minimiza o seek time nos acessos ao ficheiro em discos HDD.

21.As siglas IDE, EIDE, SCSI, SATA identificam tecnologias utilizadas em:

- a. **Controladores de disco.**
- b. Microprocessadores.
- c. Módulos de memória.

22.Qual a vantagem do seguinte esquema de indexação para a localização de blocos de ficheiros em disco?



- a. Gastar apenas espaço de disco com os dados.
- b. É muito eficiente para ficheiros pequenos.
- c. Minimiza os movimentos da cabeça de leitura.

23. A priori, uma desvantagem das técnicas de segmentação relativamente a técnicas que mapeiam processos completos em zonas contíguas de memória consiste no facto da tradução de endereços virtuais em endereços físicos:

- a. Envolver a consulta de tabelas em memória.
- b. Não poder ser feita pelo hardware (MMU).
- c. As MMUs respetivas serem muito complexas.

24. Para cada processo, a Per-Process Open File Table (PPOFT) mantém informação sobre os ficheiros por ele abertos, porque há informação dos ficheiros específica aos processos, por exemplo:

- a. A localização do cursor de leitura/escrita.
- b. As permissões de leitura, escrita e execução.
- c. A localização dos blocos respetivos em disco.

25. Devido a características de implementação das memórias NAND, o acesso a dados em discos de estado sólido (SSD) é uniforme:

- a. Para operação de escrita apenas.
- b. Para operações de leitura e de escrita.
- c. Para operações de leitura apenas.

26. A estrutura de dados utilizada para organizar a informação no Unix File System (UFS) e afins é:

- a. Uma árvore com diretórios nos nós e com ficheiros nas folhas.

- b. Uma lista ligada de ficheiros simples, sem diretórios.
- c. Um grafo dirigido acíclico com diretórios nos nós e ficheiros nas folhas.

27.No CPU, a componente designada por “Memory Management Unit” (MMU) é responsável por:

- a. Detetar acessos indevidos a zonas de memória e transferir controlo para o sistema operativo.
- b. Atribuir espaço aos processos que aguardam execução mantendo a integridade da memória física.
- c. Garantir a integridade da memória e traduzir endereços virtuais em endereços na memória física.

28.A técnica da segmentação:

- a. Preserva a localidade dos dados mas não das instruções de um programa.
- b. Em geral, não preserva a localidade dos dados nem das instruções.
- c. Preserva a localidade dos dados e das instruções de um programa.

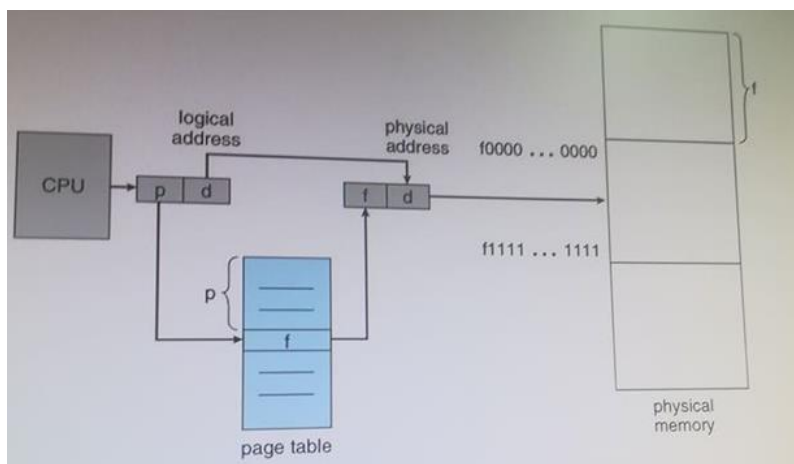
29.A utilização de blocos contíguos para guardar o conteúdo de um ficheiro:

- a. Apresenta a pior latência para acessos aleatórios no ficheiro.
- b. Simplifica a criação de ficheiros grandes e expansíveis.
- c. Minimiza o seek time nos acessos ao ficheiro em discos HDD.

30.Um sistema de ficheiros virtual (VFS) é:

- a. É um sistema de ficheiros desenhado especificamente para ser utilizado em máquinas virtuais, suportando vários sistemas operativos.
- b. Uma abstração do sistema operativo que permite oferecer aos utilizadores uma API uniforme para operações sobre o sistema de ficheiros.
- c. É um sistema de ficheiros implementado inteiramente em memória e que fornece ao utilizador a ilusão de um sistema de ficheiros em disco.

31. Na imagem da MMU que se segue, não é feita nenhuma verificação do “offset” do endereço para garantir que este cai sempre dentro da página porque:



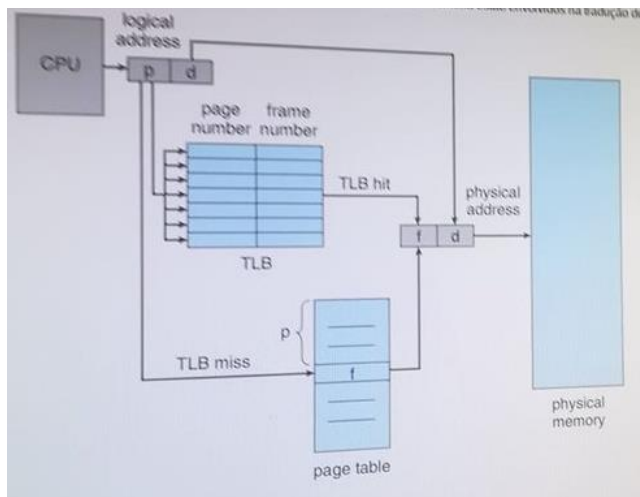
- a. Essa verificação é feita por software, pelo sistema operativo, e não pelo hardware da MMU.

- b. O hardware de tradução de endereços fica mais rápido mesmo que possa gerar erros ocasionais.
- c. A divisão em bits dos endereços virtuais garante que “offset” é inferior ao tamanho duma página.

32. Numa arquitetura de 32 bits, um sistema operativo com páginas de 4KBytes produz tabelas de páginas com quantas entradas?

- a. 2^{20} .
- b. 2^{30} .
- c. 2^{12} .

33. A imagem que se segue representa uma MMU. Quantos acessos à memória estão envolvidos na tradução de um endereço de memória virtual quando: 1) o TLB tem bit, 2) caso contrário:



- a. 1 e 2.
- b. 2 e 2.
- c. 2 e 1.

34. O seguinte programa recebe o nome de dois ficheiros como argumentos e:


```

/* includes & defines */

int main(int argc, char* argv[]) {
    /* check arguments */

    FILE* fp1 = fopen(argv[1], "w");
    if(fp1 == NULL) { /* error action */ }

    FILE* fp2 = fopen(argv[2], "r");
    if(fp2 == NULL) { /* error action */ }

    int bytes_read;
    char buffer[BUFFER_SIZE];
    while( (bytes_read = fread(buffer, 1, BUFFER_SIZE, fp2)) != 0)
        fwrite(buffer, 1, bytes_read, fp1);

    fclose(fp2);
    fclose(fp1);

    return EXIT_SUCCESS;
}

```

- a. Concatena o conteúdo do seguinte com o do primeiro.
- b. Substitui o conteúdo do primeiro pelo do segundo.
- c. Substitui o conteúdo do segundo pelo do primeiro.

35.As “pipes” são um exemplo de mecanismo de comunicação entre processos suportados pelo “kernel” do UNIX. A troca de informação é:

- a. Unidirecional, entre processos pai-filho.
- b. Unidirecional, entre quaisquer processos.
- c. Bidirecional, entre dois quaisquer processos.

36.Quantos identificadores de processos são imprimidos pelo programa que se segue?

```

/* includes & defines */

int main() {
    fork();
    fork();
    fork();
    fork();
    fork();
    printf("%d\n", getpid());
    return 0;
}

```

- a. 32.
- b. 5.
- c. 6.

37.O “Process Control Block” (PCB) é:

- a. Para além da fila “ready”, a outra fila de processos que esperam por tempo de microprocessador.

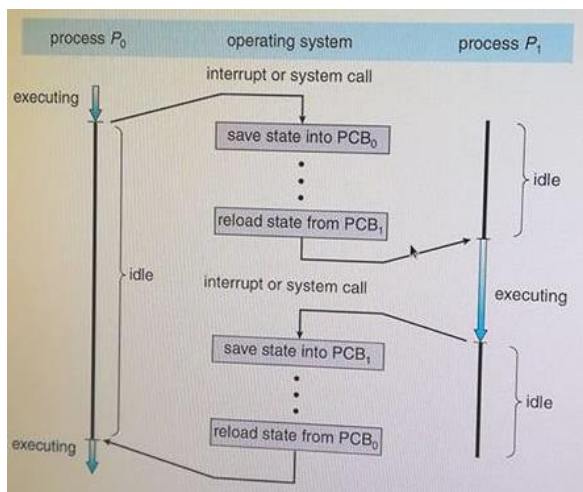
- b. O bloco de código do sistema operativo responsável pelo controlo da execução dos processos.
- c. Uma estrutura de dados onde é mantida informação sobre o estado de execução de um processo.

38. O tempo médio de espera para a seguinte configuração de processos (P1-P2-P3-P4) na fila “ready”, com o algoritmo “Round-Robin” (RR) e um quantum de 5 é:

Process	Burst-Time
P1	21
P2	6
P3	11
P4	3

- a. 19.25
- b. 12.74
- c. 21.33

39. A imagem que se segue representa um evento fundamental num algoritmo de escalonamento “pre-emptive”. Qual?



- a. Um “context switch”.
- b. Um “backup” do sistema.
- c. Uma operação de I/O.

40. Qual a relação entre os sistemas operativos iOS e Android com o UNIX?

- a. Não existe nenhuma relação entre os dois sistemas operativos e o UNIX.
- b. O iOS descende do BSD UNIX via MACOS X, o Android foi desenhado de raiz.
- c. Ambos descendem de sistemas UNIX, o BSD UNIX e o Linux, respectivamente.

41. Uma “Application’s Programmer interface” (API) é:

- a. Uma interface gráfica oferecida por alguns sistemas operativos que proporciona aos programadores um ambiente mais amigável de desenvolvimento.
- b. Um conjunto de exemplos de aplicações fornecidas com o sistema operativo e que facilitam a escrita de novas aplicações pelos programadores.

- c. Uma especificação de um conjunto de funções disponíveis para o programador, incluindo os nomes, tipos dos parâmetros e tipos dos valores de retorno.

42. Tipicamente, um processo executa:

- a. Alternando entre “bursts” de microprocessador e de I/O.
- b. Todo o I/O no início, seguindo de “burst” de microprocessador.
- c. Com um “burst” de microprocessador, o I/O é todo no fim.

43. A implementação de algoritmo de escalonamento “pre-emptive” obrigou a inovações específicas no hardware e software, por exemplo:

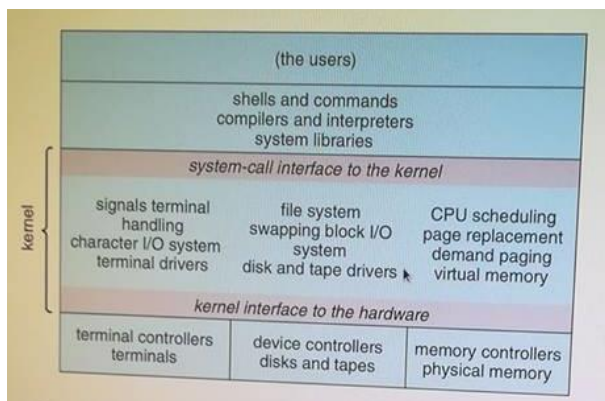
- a. Técnicas como o “Direct Memory Access” para transferir blocos grandes de dados.
- b. Inibição de interrupções durante a execução de porções críticas de código.
- c. A utilização de espaços de endereçamento individuais para os processos.

44. O seguinte programa:

```
/* includes & defines */
int main(int argc, char* argv[]) {
    char* args[0];
    int i;
    for(i = 0; i < argc - 1; i++)
        args[i] = argv[i+1];
    args[i] = NULL;
    execvp(args[0], args);
    return 0;
}
```

- a. Executa o programa dado em argv[0] com os argumentos em argv[1], etc.
- b. Falha sempre na execução da chamada ao sistema “execvp” e retorna 0.
- c. Executa o programa dado em argv[1] com os argumentos em argv[2], etc.

45. Na figura seguinte, que representa a estrutura de um sistema operativo UNIX, como se faz a interação com o “kernel”?



- a. Um utilizador nunca interage diretamente com o “kernel”.
- b. Através de funções designadas por “system calls”.

- c. Através de bibliotecas como a “Standard C Library”.

46.O “Basic Input/Output System” (BIOS) de um computador é:

- a. A parte do sistema operativo que lida com toda a interação com os dispositivos de I/O.
- b. Um programa que carrega o “kernel” do sistema operativo para a memória no arranque do computador.
- c. A componente do sistema operativo que lida apenas com dispositivos de I/O mais simples.

47.Um sistema operativo que executa, num único processador, vários processos longos e computacionalmente intensivos sem interrupções:

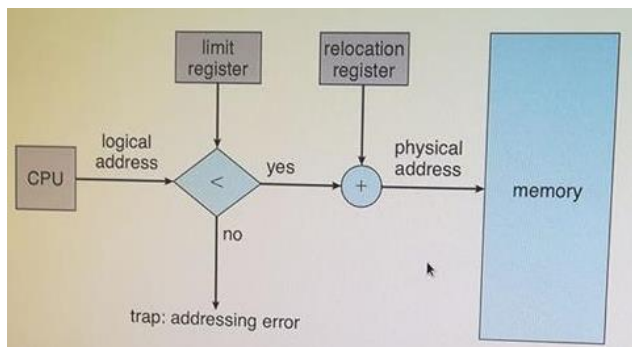
- a. Não é bom para “multitasking” nem para “multiprogramming”.
- b. É bom para “multiprogramming” mas não para “multitasking”.
- a. É bom para “multitasking” mas não para “multiprogramming”.

48.O tempo médio de espera na fila para a seguinte configuração de processos, com o algoritmo “Shortest Remaining Time First” (SRTF) é:

Process	Arrival – Time	Burst – Time
P1	0	8
P2	1	4
P3	2	9
P4	3	5

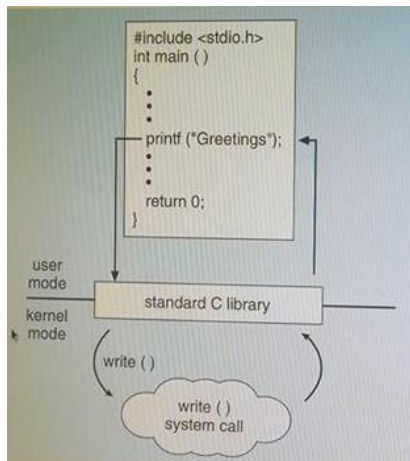
- a. 7.25
- b. 6.50
- c. 9.35

49.A “Memory Management Unit” (MMU) representada na figura seguinte não é adequada para sistemas operativos “multitasking” porque:



- a. Implica espaços de endereçamento de processos colocados num único bloco de memória física.
- b. O impacto da tradução dos endereços virtuais em endereços físicos no desempenho é enorme.
- c. Implica espaços de endereçamento de processos dispersos por vários blocos de memória física.

50.A figura seguinte representa conceptualmente o funcionamento da chamada “printf” da “Standard C Library”. A relação entre o “printf” e a chamada “write” é:



- a. O “printf” usa buffers em memória para escrever temporariamente e quando enchem chama o “write”.
- b. O “printf” invoca a chamada ao sistema “write” apenas quando o processo corrente termina.
- c. Cada invocação da função “printf” despoleta de imediato uma chamada ao sistema “write”.

51. Num “context switch”, o sistema operativo:

- a. Troca o processo corrente pelo processo seguinte sem guardar ou carregar qualquer tipo de informação contida nos registos ou PCBs.
- b. Guarda os valores dos registos no PCB do processo que sai e carrega os registos com os valores guardados no PCB do processo que entra.
- c. Guarda os valores dos registos no PCB do processo que entra e carrega os registos com os valores guardados no PCB do processo que sai.

52. O “Completely Fair Scheduler” (CFS), utilizado no “kernel” do Linux organiza os processos:

- a. Em filas simples.
- b. Em filas múltiplas.
- c. Numa árvore binária.

53. Se um processo P termina sem esperar pelo término de um processo filho Q:

- a. Q torna-se um processo “zombie”, sendo eliminado pelo processo “init”.
- b. Q não sofre qualquer alteração no seu estado e continua a executar.
- c. Q torna-se um processo órfão, sendo adotado pelo processo “init”.

54. Tipicamente, os algoritmos “Earliest Deadline First” (EDF) e/ou “Rate Monotonic Scheduling” (RMS) são utilizados em sistemas operativos que executam processos associados a tarefas:

- a. Intensivas.
- b. Interativas.
- c. De tempo real.

55. Com as novas arquiteturas de microprocessadores (e respetivas MMUs) de 64 bits:

- a. As novas MMUs apresentam o mesmo nível de suporte para a segmentação e para a paginação.

- b. O suporte para a paginação ao nível do hardware tem desaparecido e a segmentação é preferida.
- c. O suporte para a segmentação ao nível do hardware tem desaparecido e a paginação é preferida.

56. Um sistema operativo é um programa que:

- a. Verifica periodicamente o estado do hardware e gera avisos para a substituição de componentes.
- b. Permite o arranque do computador (“bootstrap”) e depois transfere o controlo para o utilizador.
- c. Corre permanentemente num computador e gere os recursos de hardware para os utilizadores.

57. Numa arquitetura de 32 bits, um sistema operativo com páginas de 2KBytes produz tabelas de páginas com quantas entradas?

- a. 2^{21} .
- b. 2^{11} .
- c. 2^{30} .

58. A função do programa “loader” do sistema operativo é:

- a. Transferir um binário executável para memória transformando-o num processo com espaço de endereçamento e PCB.
- b. Identificar funções sem código binário associado, localizá-las em bibliotecas e gerar um binário executável.
- c. Gerar código binário executável para uma arquitetura de microprocessadores a partir de código em linguagem c.