Fundamentos de Sistemas de Operação
2º Teste, 19 de Dezembro de 2022

NOME DO ESTUDANTE: _____ Nº: _____

A duração do teste é 1h45 (incluindo a tolerância). Nas questões de escolha múltipla, as respostas erradas têm uma cotação negativa correspondente a 20% da classificação da questão. Por exemplo, se errar a resposta a uma questão cotada para 1.0 valor, terá uma cotação de –0,2 valores nessa questão. A classificação total das questões de escolha múltipla pode, portanto, ser negativa.

As questões de escolha múltipla devem ser respondidas na folha própria para o efeito.
Para anular uma resposta coloque uma cruz por cima e pinte a nova resposta (✗ ○ ○ ● ○).
Para reativar uma resposta previamente anulada faça um círculo à volta da resposta a reativar (● ○ ○ ✗ ○).

As questões de desenvolvimento devem ser respondidas no próprio enunciado.

As páginas 1 e 2 podem ser usadas para **RASCUNHO**

Algumas potências de 2

2^0= 1 2^1= 2 2^2= 4 2^3= 8 2^4= 16 2^5= 32 2^6= 64 2^7= 128 2^8= 256 2^9= 512 21^{10}= 1024 (1K)
2^{11}= 2048 (2K) 2^{12}= 4196 (4K) 2^{13}= 8192 (8K) 2^{14}= 16384 (16K) 2^{15}= 32768 (32K)
2^{16}= 65536 (64K) 2^{17}= 131072 (128K) 2^{18}= 262144 (256K) 2^{19}= 524288 (512K)
21^{20}= 1048576 (1024K, 1M)

Fundamentos de Sistemas de Operação
2º Teste, 19 de Dezembro de 2022

QUESTÕES DE ESCOLHA MÚLTIPLA — VERSÃO A

1) Seja SF um sistema de ficheiros idêntico ao estudado nas aulas (e no trabalho) que organiza os **B blocos** disponíveis para armazenar os dados dos ficheiros,em **C clusters**, e seja **P o bitmap** utilizado para os gerir; considere um disco com 1200 blocos úteis para dados e que, para efeitos desta questão, a dimensão de um cluster é 2. Quantos **blocos** são necessários reservar para a zona do **bitmap**?

a) 1

b) 2

c) ½

d) nenhuma das opções anteriores é uma resposta correcta, pois é necessário saber a dimensão total do disco

e) nenhuma das opções anteriores é uma resposta correcta

2) Seja SF um sistema de ficheiros idêntico ao estudado nas aulas (e no trabalho) que organiza os **B blocos** disponíveis para armazenar os dados dos ficheiros,em **C clusters**. Contudo, para esta questão, considere que o SF usa uma atribuição em **lista ligada** para os dados dos ficheiros. Considere um disco com 10000 blocos úteis para dados formatado de forma a que a dimensão de um cluster seja 4. Qual é a dimensão mínima (em bytes) do apontador que vai ligar os diversos clusters de um dado ficheiro?

a) 1

b) 2

c) 3

d) 4

e) nenhuma das opções anteriores é uma resposta correcta, pois é necessário saber a dimensão total do disco

3) Assinale **a ou as** frases que indicam definições e/ou situações que são verdadeiras (Nota: MV = máquina virtual)

a) um processo executa-se sobre uma MV criada pelo SO, que o isola relativamente aos outros processos que se executam sobre o mesmo SO, mas partilha os recursos e serviços que este (SO) gere/disponibiliza

b) um SO executa-se sobre uma MV criada por um hipervisor, que a isola relativamente às outras MVs que se executam sobre o mesmo hipervisor, mas partilha os recursos e serviços que este (hipervisor) gere/ /disponibiliza;

c) um programa Java executa-se sobre uma MV implementada por uma JVM (Java *Virtual Machine*) que é um processo executado sobre uma MV criada pelo SO, que o isola relativamente aos outros processos que se executam sobre o mesmo SO, mas partilha os recursos e serviços que este (SO) gere/disponibiliza

d) um *container* é um processo que é executado sobre uma MV criada pelo SO, que o isola relativamente aos outros processos que se executam sobre o mesmo SO, mas partilha os recursos e serviços que este (SO) gere/disponibiliza

e) um *container* é um SO (dito *guest*, ou hospedado) que é executado sobre uma MV criada pelo SO (dito *host*, ou hospedeiro), que o isola relativamente aos outros *containers* que se executam sobre o mesmo SO hospedeiro, mas partilha os recursos e serviços que este (SO) gere/disponibiliza

4) Seja SF um sistema de ficheiros idêntico ao estudado nas aulas (e no trabalho) que organiza os **`B` blocos** disponíveis para armazenar os dados dos ficheiros, usando **`C` clusters**. Para esta questão, considere que os i-nodes do SF usam uma atribuição **indexada** com apontadores de **32 bits** sendo que os i-nodes têm um único apontador, de tipo N-indirecto (onde N será o nº de níveis, e.g., duplamente-indirecto, ou triplamente-indirecto, etc.), também de **32 bits** para referenciar ou os dados dos ficheiros, ou os outros índices. Considere um disco formatado de forma a que, tanto a dimensão de um cluster, como a de um índice, é 2 (blocos). Quantos níveis terá a árvore de índices quando a dimensão do ficheiro for 1MB?

a) 1

b) 2

c) 3

d) 4

e) 5

5) Considere um sistema de ficheiros de utilização "real" como, por exemplo, o **ext2** do Linux (e não um simples SF que serve para estudo). Durante a utilização de um computador com disco formatado nesse SF um problema eléctrico grave na rede provocou uma destruição da informação existente no superbloco em uso naquele instante. O resultado é um disco

a) irrecuperável, pois não é possível saber nada sobre os restantes metadados desse disco.

b) irrecuperável, pois não é possível saber nada sobre a restante informação nesse disco

c) recuperável, basta re-arrancar o sistema que o mecanismo de *journaling* recupera o superbloco a partir do *log* das transacções em curso

d) recuperável, basta re-arrancar o sistema que o programa **`fsck`** (file system checker) recupera o superbloco a partir das informações existentes noutras cópias do superbloco

e) recuperável, basta re-arrancar o sistema que o programa **`fsck`** (file system checker) recupera o superbloco a partir das informações existentes nas zonas de bitmaps e i-nodes.

6) Num SF "simples" como o estudado nas aulas (e no livro recomendado) a operação de **`write()`** quando este faz aumentar a dimensão de um ficheiro, as estruturas de dados do SF que são alteradas são

a) a directoria, o i-node, o(s) blocos (ou *clusters*) de dados que serão escritos e, eventualmente, o bitmap de blocos (ou *clusters*) de dados se for necessário atribuir ("alocar") um (ou mais) novo(s) bloco(s) (ou *cluster(s)*) de dados

b) o superbloco, o bitmap de i-nodes, o i-node, o(s) blocos (ou *clusters*) de dados que serão escritos e, eventualmente, o bitmap de blocos (ou *clusters*) de dados, se for necessário atribuir ("alocar") um (ou mais) novo(s) bloco(s) (ou *cluster(s)*) de dados

c) o superbloco, o i-node, o(s) blocos (ou *clusters*) de dados que serão escritos e, eventualmente, o bitmap de blocos (ou *clusters*) de dados se for necessário atribuir ("alocar") um (ou mais) novo(s) bloco(s) (ou *cluster(s)*) de dados

d) o bitmap de i-nodes, o i-node, o(s) blocos (ou *clusters*) de dados que serão escritos e, eventualmente, o bitmap de blocos (ou *clusters*) de dados, se for necessário atribuir ("alocar") um (ou mais) novo(s) bloco(s) (ou *cluster(s)*) de dados

e) o i-node, o(s) blocos (ou *clusters*) de dados que serão escritos e, eventualmente, o bitmap de blocos (ou *clusters*) de dados se for necessário atribuir ("alocar") um (ou mais) novo(s) bloco(s) (ou *cluster(s)*) de dados

7) Num computador com um processador do tipo x86 que corre um SO que suporta "memória virtual por paginação-a-pedido" (*demand paging*), qual das seguintes afirmações sobre a dimensão do espaço de endereçamento (EE) de um processo é a mais correcta?

a) Pode exceder a dimensão da RAM instalada no computador

b) Pode exceder a dimensão da RAM instalada no computador, sendo limitada pela dimensão dos registos **eax**, **ebx**, **ecx**, **edx**

c) Não pode exceder a dimensão da RAM instalada no computador

d) Não pode exceder a dimensão da RAM instalada no computador e é limitada pela dimensão dos registos **eip** e **esp**

e) Pode exceder a dimensão da RAM instalada no computador, sendo limitada pela dimensão dos registos **eip** e **esp**

8) Num computador com um processador do tipo x86 que corre um SO que suporta "memória virtual por paginação-a-pedido" (*demand paging*), o número de páginas **válidas** de um processo que suporta tarefas típicas do dia-a-dia de um utilizador que escreve relatórios e prepara apresentações (e.g., Word, Power-Point – ou aplicações "simlares" em Linux, como o OpenOffice Write ou Calc)

a) é muito reduzido, face ao número total de páginas que pode existir no espaço de endereçamento, o que se traduz em mapas ou tabelas de páginas (*page tables*) muito pouco preenchidas (ditas esparsas) que, por isso, são implementadas recorrendo a indexação com múltiplos níveis

b) é muito reduzido, face ao número total de páginas que pode existir no espaço de endereçamento, o que se traduz em mapas ou tabelas de páginas (*page tables*) muito pouco preenchidas (ditas esparsas) que, por isso, são implementadas recorrendo a técnicas de compressão (de tipo ZIP)

c) é muito reduzido, face ao número total de páginas que pode existir no espaço de endereçamento, o que se traduz em mapas ou tabelas de páginas (*page tables*) muito pouco preenchidas (ditas esparsas) que, por isso, são implementadas recorrendo tabelas "completas"/lineares

d) é elevado, e próximo do número total de páginas que pode existir no espaço de endereçamento, o que se traduz em mapas ou tabelas de páginas (*page tables*) muito preenchidas que, por isso, são implementadas recorrendo a indexação com múltiplos níveis

e) é elevado, e próximo do número total de páginas que pode existir no espaço de endereçamento, o que se traduz em mapas ou tabelas de páginas (*page tables*) muito preenchidas que, por isso, são implementadas recorrendo a técnicas de compressão (de tipo ZIP)

9) Assinale qual das seguintes afirmações é uma correcta caracterização da técnica de atribuição de blocos a um ficheiro descrita: (Nota: um acesso directo envolve um **`lseek()`** seguido de **`read()`** ou **`write()`**)

a) a atribuição contígua apresenta um acesso directo muito lento

b) a atribuição ligada apresenta dificuldades no crescimento dos ficheiros

c) a atribuição indexada apresenta um acesso directo muito lento

d) a atribuição contígua conduz à maior rapidez possível em acesso directo

e) a atribuição indexada conduz à maior rapidez possível em acesso directo

10) Quando uma página (válida) de um processo está carregada (presente) em RAM e o processo tenta aceder a essa página usando um endereço **`E`** (pertencente a essa página), qual das seguintes afirmações é verdadeira?

a) o endereço virtual **`E`** é transformado em real pelo processador sem auxílio do TLB, pois se a página está em RAM a tradução já foi feita anteriormente, e o processador guardou-a em *cache*

b) o endereço virtual **`E`** é transformado em real pelo processador, sem auxílio do TLB se o processador ainda tiver a tradução página/*frame* em *cache*, ou com auxílio do TLB em caso contrário

c) o endereço virtual **`E`** é transformado em real pelo TLB, que usa a tradução página/*frame* existente em *cache*, pois se a página está em RAM a tradução já foi feita anteriormente e está garantidamente na *cache*

d) o endereço virtual **`E`** é transformado em real pelo TLB que pode, ou não, ter a tradução página/*frame* em *cache* e, consequentemente pode, ou não, ter de gerir um *miss* durante o processo de tradução

e) o endereço virtual **`E`** é transformado em real pelo processador recorrendo unicamente ao SO

11) Um processo recebe um sinal de falta de segmentação (*segmentation fault*, **`SIGSEGV`**) quando

a) tenta aceder a um ficheiro (e.g., usando **`read()`** ou **`write()`**) sendo que o descritor do ficheiro (*file descriptor*) é inválido

b) tenta usar uma instrução privilegiada sem que o processo seja *root*

c) tenta efectuar uma operação de acesso a um endereço (de memória) que pertence a uma página para a qual o processo não tem as necessárias permissões

d) tenta efectuar uma operação de acesso a um endereço (de memória) que pertence a uma página que não está carregada em RAM

e) tenta aceder a um ficheiro usando **`mmap()`** sendo que o descritor do ficheiro (*file descriptor*) é inválido

12) Uma rotina de tratamento de sinais (*signal handler*) é uma rotina

a) do núcleo (*kernel*) do SO, invocada por este (SO) quando um processo gera um sinal

b) do programa que trata um sinal específico (com excepção dos que por definição não podem ser tratados) e é activada pelo *kernel* quando o processo (que está a executar o referido programa) recebe esse sinal

c) do programa que trata o sinal *segmentation fault* (**`SIGSEGV`**) e que é activada pelo TLB quando este detecta um acesso ilegal à memória

d) do programa que trata todos os sinais (com excepção dos que por definição não podem ser tratados) e é activada sob a forma de *thread* quando o processo (que está a executar o referido programa) gera um desses sinais

e) do programa que trata todos os sinais (com excepção dos que por definição não podem ser tratados) e é activada pela função **`signal()`** da biblioteca do C quando o processo (que está a executar o referido programa) recebe um desses sinais

13) O que caracteriza um escalonador que usa o algoritmo MLFQ (*Multi-Level Feedback Queue*) é

a) não usar fatias de tempo

b) favorecer a execução dos processos CPU-bound

c) favorecer a execução dos jobs mais curtos (SJF)

d) escalonar os processos preservando a sua antiguidade (exibindo um carácter FIFO)

e) nenhuma das anteriores

14) Depois de uma falha de energia ou de um *crash* SO, num computador com um volume formatado com um determinado sistema de ficheiros (SF) e em uso (montado) exclusivamente para **leitura**, a consistência do SF

a) **tem** de ser verificada

b) **não** tem de ser verificada

c) só tem de ser verificada se se tratar de um SF **com** journaling

d) só tem de ser verificada se se tratar de um SF **sem** journaling

e) só tem de ser verificada se se tratar de um SF nativo do SO em causa (por exemplo, **`ext2`** num SO Linux ou NTFS num SO Windows)

15) Num disco com **`B`** blocos, formatado para conter um sistema de ficheiros que suporta atribuição (de blocos de dados) baseada unicamente num vector com **`N`** apontadores directos de dimensão A_p **bytes**, e em que cada apontador aponta para um *cluster* de 8 blocos, a dimensão mínima para um **apontador**, A_p, é dada por [Nota: **`B`** e **`N`** são potências de 2)

a) $A_p = ceil(\log_2(N)/B) \ / \ 8$

b) $A_p = ceil(\log_2(B)/8)$

c) $A_p = ceil(\ log2(B/8) \ / \ 8)$

d) $A_p = ceil(\log_2(N/8) \ / \ 8)$

e) $A_p = (2^8)*ceil(\log_2(B/8))$

16) Suponha um utilizador que, por intermédio de uma aplicação dessas que são típicas dos sites de compras on-line, está a "encher" o seu "carrinho de compras" com produtos que vai pesquisando, seleccionando, e finalmente comprando (i.e., colocando no "carrinho"). Do ponto de vista do utilizador (ou, se quiser, do programador ou do gestor do sistema onde corre a aplicação, se estes se colocarem "na pele do utilizador") qual o critério mais importante para caracterizar o desempenho ("performance") do sistema?

a) tempo de resposta

b) débito (ou *throughtput*)

c) *turnaround*

d) latência

e) nenhum dos anteriores

17) Um *device driver* é

a) um módulo incluído na biblioteca **`stdio`** que permite ao programador actuar sobre periféricos sem necessidade de recorrer ao modo supervisor (ou *system*) do CPU

b) um módulo do núcleo do SO que é responsável pela gestão (controle) de um periférico e geralmente acedido pelas camadas "superiores" do SO (e.g., sistema de ficheiros)

c) o *firmware* que vem instalado num periférico e que controla o seu funcionamento

d) o *firmware* que vem instalado num computador e constitui parte do que se chama a **BIOS** e que controla o seu funcionamento

e) nenhum dos anteriores

18) Num sistema computacional, a autenticação é o processo

a) que permite ao SO verificar se uma entidade interna de um sistema (e.g., um processo ou *thread*) pode manipular um recurso (ficheiro ou página de memória)

b) que permite ao SO verificar se uma entidade interna de um sistema (e.g., um processo ou *thread*) pode manipular um recurso protegido por uma lista de controle de acessos (user,group,others)(rwx)

c) designado vulgarmente por *login*, no qual um utilizador fornece um par de valores (*username*, *password*) que, se determinado como correcto, faz com que esse utilizador tenha acesso aos recursos do sistema

d) pelo qual uma entidade activa (pessoa ou programa) externa tem de passar para conseguir ter acesso ao referido sistema computacional e consiste em aceitar um desafio (e.g., *username*, *password*, leitura biométrica, chave física, chave digital) que permita provar que a entidade é realmente quem diz ser

e) nenhuma das respostas anteriores é correcta

19) Os parâmetros que contribuem de forma **relevante** para o desempenho de uma unidade de disco magnético (HDD – Hard Disk Drive) são:

a) a velocidade de rotação, os tempos de acesso das cabeças (pista-a-pista, varrimento total ou *full stroke*, e varrimento médio ou *average seek*), e a velocidade de transferência entre a unidade de disco e a memória (RAM)

b) a velocidade de rotação, os tempos de acesso das cabeças (pista-a-pista, varrimento total ou *full stroke*, e varrimento médio ou *average seek*), velocidade de transferência entre a unidade de disco e a memória (RAM) e a capacidade total do disco

c) a velocidade de rotação, os tempos de acesso das cabeças (pista-a-pista, varrimento total ou *full stroke*, e varrimento médio ou *average seek*), o tempo de transferência entre a unidade de disco e a memória (RAM), o número de pistas e o número de sectores

d) a velocidade de rotação, os tempos de acesso das cabeças (pista-a-pista, varrimento total ou *full stroke*, e varrimento médio ou *average seek*), o tempo de transferência entre a unidade de disco e a memória (RAM), o número de pistas e o número de sectores e a capacidade de cada pista

e) capacidade do disco, da *cache*, e velocidade do *bus* que liga a unidade de disco à memória (RAM)

20) Um processo quer abrir um ficheiro cuja máscara de permissões é `000` (exibidas no *output* do **`ls -l`** pela "máscara" `----------`) e depois efectuar um `read()`. Pode fazê-lo

a) porque qualquer processo pode abrir e ler o ficheiro

b) desde que a página onde reside a chamada de sistema `read()` tenha o atributo "read" activo

c) se o UID do processo corresponde ao UID do dono (*owner*) do ficheiro ou a um UID pertencente ao mesmo grupo a que pertence o dono do ficheiro – e isto qualquer que seja o dono.

d) o processo estiver a correr em modo supervisor (ou *kernel*)

e) unicamente se o UID do processo for root (i.e., zero)

Fundamentos de Sistemas de Operação
2º Teste, 19 de Dezembro de 2022

QUESTÕES DE DESENVOLVIMENTO — VERSÃO A

D1) Considere o fragmento de programa (fonte) abaixo (Fig 1.1) que é executado num computador "imaginário" com um SO que usa paginação-a-pedido. O CPU tem registos gerais de **32 bits** mas usa endereços (físicos e lógicos – ou virtuais) com apenas **10 bits**, e páginas de **8 bytes**. O espaço de endereçamento (virtual, Fig. 1.2), o mapa (ou tabela) de páginas (Fig. 1.3), e a RAM (Fig. 1.4) estão também desenhados.

```
…
int v[MAX];
for (i= 0; i < MAX; i++) v[i]= i;
…
```

Fig 1.1

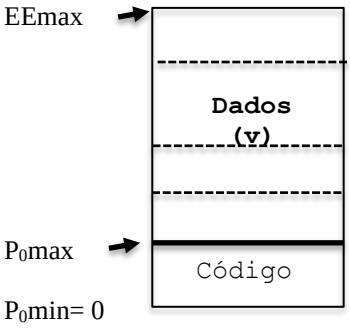


Fig. 1.2

Página	Frame
0	0
1	NP
…	…

Fig. 1.3

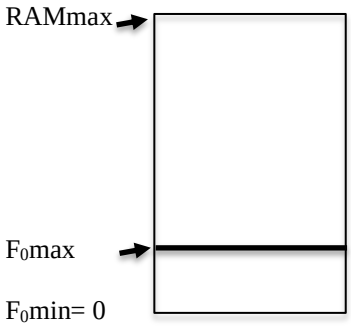


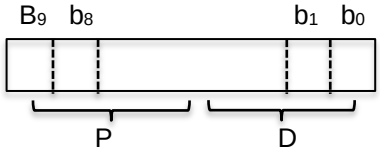
Fig. 1.4

Notas adicionais (**Ier com atenção**):
i) a variável i foi "alocada" num registo, não está em memória; v não está no *stack*, mas é global
ii) "Faça de conta" que o SO não ocupa memória.
iii) A indicação NP indica que a página em causa é válida mas não está presente (NP) em memória.
iv) EEmax e RAMmax representam os maiores endereços possíveis para, respectivamente, o Espaço de Endereçamento virtual (EEmax) e físico (RAMmax).
v) Analogamente, P_0max e F_0max representam os maiores deslocamentos possíveis para, respectivamente, uma página (neste exemplo, a página zero) e uma *frame* (também, neste caso, a zero).
vi) Só há uma página de código, P_0, que é imediatamente seguida pelas páginas de dados, P_1. … P_N ; não há *stack* nem *heap*.

a) Calcule os seguintes valores
EEmax= RAMmax=

P_0max= F_0max=

b) Decomponha um endereço virtual, de 10 bits, em "Nº de página" (P) e "deslocamento" (D) indicando na figura abaixo, quantos bits têm P e D



P= ___ bits D= ___ bits

c) Considere MAX = 32. Calcule P_{vTotal}, o número de páginas ocupado por v[MAX] e P_{fTotal}, o número de faltas de página sofridas pelo programa até que se complete o ciclo **`for`**. [Recorde que se trata de uma arquitectura com registos de **32 bits** e v é um vector de **int**eiros] e determine se há sempre *frames* livres ou se há necessidade de fazer uma substituição de página (*page replacement*) – e, se houver, indique uma vítima provável. Explane o seu raciocínio para calcular os 3 valores

P_{vTotal} = P_{fTotal} = $P_{vitimas}$ =

D2) Preencha nas frases os espaços abaixo com os **números** que indicam o termo mais apropriado.

Exemplo: Cá por Lisboa tem __2___ muito.

Termos a usar: (1) nevado (2) chovido

Frases a completar:

Para que um disco possa ser usado, e se consiga aceder aos ficheiros nele existentes, é preciso que seja primeiro ____. Esta operação carrega em memória o ____ e determina quais as zonas onde começam e acabam as estruturas de dados necessárias para a gestão de ficheiros; algumas destas são: ____, ____, e ____.

Após uma falta (por falha de energia, ou *crash* do SO) é necessário verificar a ____ da informação do sistema de ficheiros; tal operação é (geralmente) desencadeada automaticamente quando o disco é ____, podendo contudo ser executada manualmente. Nos sistemas de ficheiros que não usam ____ o processo de verificação pode demorar muito tempo.

Quando um processo faz um **`fork()`**, o SO cria para o filho um(a) ____ que aponta a ____ do novo processo. Esta é idêntica à do processo pai - tem o mesmo número de entradas e estas têm os mesmos atributos ("permissões") e apontam para as mesmas ____. Contudo, à medida que a execução dos dois processos, pai e filho, vai progredindo, e os processos vão alterando valores de variáveis globais, ou em zonas de ____ e/ou ____, as tabelas de páginas de ambos vão ficando diferentes uma da outra, e o mecanismo responsável por essa diferenciação é o ____, que cada vez que uma página partilhada entre os dois processos é pela primeira vez modificada, procura uma ____ livre, faz para esta uma cópia da ____ "original", já existente, a alterar, altera a entrada correspondente na ____, e finalmente promove a alteração pretendida.

Por outro lado, quando um processo faz um **`pthread_create()`**, o SO cria para a nova *thread* um(a) novo(a) ____ mas não há criação de uma nova ____, nem o mecanismo de ____ intervém quando a thread altera valores no espaço de endereçamento do processo.

Termos/números a usar (nota: um número pode ser usado mais do que uma vez, ou nenhuma):

(1) tabela de páginas (2) COW (Copy-On-Write) (3) bitmap de i-nodes (4) *journal*
(5) consistência (6) tabela de i-nodes (7) stack
(8) bitmap de dados (9) clusters (10) montado (11) task struct
(12) PCB (Process Control Block) (13) superbloco (14) *frame(s)*
(15) página (16) PTBR (Page Table Base Register) (17) heap
(18) TCB (Thread Control Block)