Exame | Instruções para a realização de exame

Sistemas Operativos | 21111

Data de Realização

Dia 18 de junho de 2024

Duração da prova (wiseflow)

120 minutos, mais 60 minutos de tolerância

Trabalho a desenvolver

Responder às questões dos Grupos I e II.

Leia estas informações e instruções na totalidade antes de iniciar a resolução da prova.

Critérios de avaliação e cotação

- As cotações são indicadas por grupo e nas próprias questões.
- As respostas às questões devem fazer sentido, ser coerentes e constituídas por palavras próprias do aluno. Não serão aceites transcrições ou traduções de livros e textos, incluindo textos de orientações de respostas de provas anteriores. As respostas que não respeitem estas condições serão classificadas com zero valores ou fortemente desvalorizadas.
- Nas questões de resposta aberta, não existem respostas únicas ou definitivas. As respostas devem ser relativamente desenvolvidas e elaboradas de modo a demonstrar quer conhecimento específico quer conhecimento geral sobre os tópicos das questões. A clareza do texto e da explicação também são levadas em conta na classificação das respostas.
- Nas questões de escrita de programas, a sua correção tem em conta critérios de proficiência e compreensibilidade do código tais como: legibilidade, indentação, estrutura, comentários e explicação geral do seu funcionamento.

Normas a respeitar

- Está a redigir a sua prova na WISEflow. A prova não será de consulta, exceto se existirem materiais ou recursos indicados pelo Professor neste enunciado.
- Deve identificar claramente, e em **bold**, o número de cada questão a que está a responder. As respostas devem ser ordenadas por ordem crescente. Sendo a identificação automática, não deve inserir uma folha de rosto antes das respostas a esta prova, pois a folha de rosto será gerada automaticamente na WISEflow.
- O texto de todas as respostas deve ser introduzido pelo processador de texto, não sendo aceites respostas escritas à mão ou por outros meios, digitalizadas e incluídas como imagens. São exceções expressões matemáticas mais complicadas, devendo ter legenda ou identificação de modo a serem referidas nos textos explicativos (opcionalmente pode recorrer a um anexo WISEflow tipo "Editor de fórmulas").
- No caso de código de programas é obrigatório a sua introdução por um anexo WISEflow tipo "Código".
- No caso de figuras e diagramas é obrigatório a sua introdução por um anexo WISEflow tipo "Desenho".
- Se fizer a prova remotamente, deve ter um comportamento em tudo semelhante à realização da prova em contexto presencial num centro de exame.
- O(a) estudante em avaliação remota deve, durante a prova online realizada através da WISEflow, seguir as seguintes instruções:
  - Não se pode levantar durante a prova, incluindo ir à casa de banho;
  - Deve procurar um lugar calmo, onde possa estar sozinho, com as costas viradas para uma parede;
  - Deve desligar o telemóvel, ou qualquer outro dispositivo informático, com o qual possa aceder à Internet;
  - No caso de se tratar de uma prova sem consulta, deve retirar todas as folhas, livros ou fotocópias de cima da mesa onde realizará a prova, exceto se autorizado pelo docente;
  - Durante a prova, não pode conversar com pessoas independentemente, do teor da conversa.
- Assim que estiver pronto(a) para submeter a prova, deve selecionar a opção “ir para entrega” que está sinalizada a verde no canto superior direito da página.

Votos de bom trabalho!

Paulo Shirley

Grupo I [12 valores]

1.1. [1.2] Os sistemas operativos utilizam abstrações como estratégia para gerir a complexidade. Indique as três principais abstrações e os respetivos recursos a elas associados.

1.2. [1.2] Explique em que consiste um interpretador de comandos (shell) e o seu modo de operação. Este é considerado como pertencente ao sistema operativo ?

1.3. [1.2] Explique o conceito de pseudoparalelismo e relacione-o com o conceito de processo.

1.4. [1.2] Explique o conceito de tarefa (thread). Qual a sua relação com o conceito de processo ?

1.5. [1.2] O que entende por uma condição de disputa (race condition) ?

1.6. [1.2] O que significa um processo encontrar-se em privação ou míngua (starvation) ?

1.7. [1.2] O conceito de espaço de endereçamento virtual unidimensional permite atribuir um espaço próprio de endereçamento para cada processo ou tarefa (thread) ? Justifique.

1.8. [1.2] Explique sucintamente o algoritmo de substituição de páginas do conjunto de trabalho (Working set page replacement algorithm).

1.9. [1.2] Para um sistema de ficheiros baseado em i-nodes, explique no que consiste e como é implementado um "hard link".

1.10. [1.2] Explique como o sistema operativo gere os processos quando um processo efectua um pedido de entrada/saída (I/O) de um dispositivo que opera por interrupção.

Grupo II [8 valores]

2.1. [3] Escreva um programa em linguagem C que crie um subprocesso e que com recurso a uma função exec() execute o comando "cp f1.txt f2.txt". O comando cp encontra-se na directoria /bin. O processo pai deve testar a ocorrência de erro na criação do processo filho assim como esperar que o processo filho termine.

2.2. [5] Escreva um programa multitarefa em linguagem C segundo a norma POSIX que determine a soma S dos elementos de um vector int x[] de dimensão int nx. O vector e a sua dimensão constituem variáveis globais ao programa e admite-se que foram devidamente inicializados.

A tarefa principal deve criar dez subtarefas em que cada uma em paralelo (ou em pseudo-paralelismo) substitui de cada vez dois elementos do vector x[] por um único correspondente à sua soma, após o que deve dar oportunidade às outras tarefas de também realizarem trabalho. Note que a dimensão nx do vector x[] diminui de uma unidade cada vez que uma tarefa opera no vector, devendo as tarefas terminar quando se atingir nx=1, caso em que x[0] contém a soma total do vector. A tarefa principal deve esperar que todas as subtarefas terminem e depois deve imprimir o resultado final e terminar.

Nota: planeie cuidadosamente como é dividido o trabalho entre as sub-tarefas e como é efectuada a sincronização e a comunicação da informação necessária à resolução do problema entre as onze tarefas.

Formulário

```
#include <stdlib.h>
int system(char *string);

#include <sys/types.h>
#include <unistd.h>
pid_t fork(void);
pid_t getpid(void);
pid_t getppid(void);

#include <unistd.h>
unsigned int sleep(unsigned int seconds);
extern char **environ;
int execl(char *path, char *arg, ...);
int execlp(char *file, char *arg, ...);
int execle(char *path, char *arg , ..., char *envp[]);
int execv(char *path, char *argv[]);
int execvp(char *file, char *argv[]);
int execve(char *path, char *argv [], char *envp[]);

#include <sys/types.h>
#include <sys/wait.h>
pid_t wait(int *status);

#include <pthread.h>
int pthread_create(pthread_t *thread, pthread_attr_t *attr,
   void *(*start_routine)(void*), void *arg);
int pthread_attr_init(pthread_attr_t *attr);
int pthread_attr_setdetachstate(pthread_attr_t *attr, int detachstate);
#define PTHREAD_CREATE_DETACHED
#define PTHREAD_CREATE_JOINABLE
int pthread_join(pthread_t thread, void **value_ptr);
int pthread_mutex_init(pthread_mutex_t *mutex,
   pthread_mutexattr_t * attr);
int pthread_mutex_lock(pthread_mutex_t *mutex);
int pthread_mutex_unlock(pthread_mutex_t *mutex);
int pthread_mutex_destroy(pthread_mutex_t *mutex);
```

FIM