

Prova Escrita (sem consulta) – Exame Recurso - Enunciado E

2018-02-15

Duração: 50 min

Nome completo: _____ N.º aluno: _____

Regime: Diurno [] Pós-laboral []

Todas as perguntas devem ser resolvidas no enunciado!

[17 valores]

- Considere que todas as perguntas deste grupo são independentes.
- **Todas as perguntas respondidas incorretamente ou de forma ambígua descontam 25% da cotação da pergunta.**
- Selecione a resposta mais completa para cada uma das seguintes questões.
- Prova sem consulta.
- É expressamente proibido o uso de telemóveis ou de qualquer outro dispositivo eletrónico.

Tabela de respostas

Escreva, de forma **legível**, no retângulo reservado para o efeito, a **letra** da opção que considera a resposta certa. Caso não pretenda responder à pergunta, escreva “X” no meio do retângulo.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17			
Estudante																		18a	18b	18c
Professor																				

1. [1 valor] A codificação UTF-8 do símbolo *Russian Ruble Sign* (₽) inicia-se pela sequência binária 11100010.

Quantos octetos (*bytes*) são necessários à codificação UTF-8 do *Russian Ruble Sign*?

- a) 1
- b) 2
- c) 3
- d) 4

2. [1 valor] O protocolo HTTP/2.....

- a) É um protocolo da camada de transporte
- b) É um protocolo da camada de rede
- c) É um protocolo da camada aplicacional
- d) Nenhuma das anteriores

3. [1 valor] O protocolo TCP...

- a) garante a ordem de entrega das mensagens
- b) apenas permite operações ponto-a-ponto
- c) é orientado à ligação
- d) todas as anteriores

4. [1 valor] Considere a variável A, declarada da seguinte forma: `int32_t A;`
O menor e maior valor suportados pela variável são, respetivamente...

- a) -2^{32} e $+2^{32}$
- b) -2^{31} e $+2^{31}$
- c) -2^{31} e $+2^{31}-1$
- d) Nenhuma das anteriores

5. [1 valor] O que ocorre quando se especifica `MSG_TRUNC` no parâmetro flags da função `recvfrom`? (protótipo: `ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags, struct sockaddr *src_addr, socklen_t *addrlen);`)...

- a) A função `recvfrom` não bloqueia, retornando imediatamente após a chamada independentemente da receção ou não de dados
- b) A função `recvfrom` só retorna quando receber a totalidade do datagrama UDP
- c) A função `recvfrom` devolve o número de bytes real do datagrama e não o número de bytes escritos no buffer de receção
- d) Nenhuma das anteriores

6. [1 valor] No protocolo TCP, a receção de um segmento do tipo ACK...

- a) é empregue pela outra extremidade comunicante para indicar a existência de dados fora de banda (*out of band*)
- b) é empregue para confirmar, pela outra extremidade comunicante, a receção de segmento(s)
- c) é empregue pela outra extremidade comunicante para indicar que a ligação foi alvo de um *hACK*, isto é, um acesso ilegal e potencialmente malicioso
- d) nenhuma das anteriores

7. [1 valor] No contexto da API `socket`, a função `select` (protótipo: `int select(int nfds, fd_set *readfds, fd_set *writefds, fd_set *exceptfds, struct timeval *timeout);`)...

- a) permite detetar eventos de leitura, escrita e exceção nos descritores especificados pela entidade chamante
- b) é usualmente uma chamada bloqueante
- c) o parâmetro *timeout* pode ser especificado como sendo `NULL`
- d) todas as anteriores

8. [1 valor] A função `freeaddrinfo` (protótipo: `void freeaddrinfo(struct addrinfo *res);`)...

- a) Destina-se a libertar os recursos alocados pela função `gethostbyaddr`
- b) Destina-se a libertar os recursos alocados pela função `getnameinfo`
- c) Destina-se a libertar os recursos alocados pela função `gethostbyname`
- d) Nenhuma das anteriores

9. [1 valor] Considere que num determinado sistema Linux, se verifica que `sizeof(char) = 1`; `sizeof(short) = 2` e `sizeof(int) = 4`. Considere ainda a seguinte declaração em linguagem C:

```
uint8_t      info[16];
```

Uma declaração equivalente é...

- a) `signed char info[16];`
 - b) `unsigned char info[16];`
 - c) `unsigned short info[8];`
 - d) nenhuma das anteriores
10. [1 valor] Considere que se pretende, em linguagem C, uma expressão que tome o valor lógico verdadeiro quando e só quando o 3º e 5º bits menos significativos de uma variável A são ambos 1, não interessando o valor dos restantes bits. A variável é declarada da seguinte forma: `uint32_t A`; Qual é a expressão apropriada?

- a) `A & (1<<2) & (1<<4)`
- b) `A | ((1<<2) ^ (1<<4))`
- c) `A & ((1<<2) | (1<<4))`
- d) Nenhuma das anteriores

11. [1 valor] Considere o seguinte código C:

```
uint16_t A = UINT16_MAX;
```

```
A++;
```

O que se pode dizer sobre o valor da variável A após a execução do código?

- a) A tem o valor de `UINT16_MAX + 1`, ou seja, 65536
 - b) A tem o valor de -1
 - c) A tem o valor de 0
 - d) Nenhuma das anteriores
12. [1 valor] O compilador GCC disponibiliza um tipo de dados inteiro com 128 bits, identificado por `__int128_t`. Considere que se declara a seguinte variável numa plataforma X86_64 que executa uma versão de 64 bits do sistema operativo LINUX.

```
__int128_t A = 0x00112233445566778899AABBCCDDEEFF;
```

Qual será o *byte* que ocorre primeiro na memória (isto é, no endereço mais baixo da memória)?

- a) *byte* 0x00 porque é o *byte* mais significativo e a plataforma X86_64 é *big endian*
 - b) *byte* 0x00 porque é o *byte* mais significativo e a plataforma X86_64 é *little endian*
 - c) *byte* 0xFF porque é o *byte* menos significativo e a plataforma X86_64 é *little endian*
 - d) *byte* 0xFF porque é o *byte* menos significativo e a plataforma X86_64 é *big endian*
13. [1 valor] A função `accept` (protótipo: `int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);`)...
- a) bloqueia o processo chamante até que seja recebido um pedido de ligação no socket UDP indicado pelo parâmetro `sockfd`
 - b) bloqueia o processo chamante até que seja recebido um pedido de ligação no socket TCP indicado pelo parâmetro `sockfd`
 - c) devolve um inteiro que corresponde ao socket de escuta, com o qual deverá ser novamente chamada a função `accept`
 - d) nenhuma das anteriores

14. [1 valor] No protocolo HTTP, o código de estado 500 é empregue...
- a) Para que o servidor assinale que existe uma anomalia, inconformidade ou inexistência do recurso pretendido pelo cliente
 - b) Para que o cliente assinale que existe uma anomalia, inconformidade ou inexistência do recurso recebido do servidor
 - c) Para indicar que o processamento do pedido do cliente falhou no servidor por causa imputável ao próprio servidor
 - d) Nenhuma das anteriores
15. [1 valor] Quando na chamada da função `socket` se especifica `AF_INET6` para o domínio e `SOCK_STREAM` para o tipo, está-se a criar...
- a) Um socket UDP IPv4
 - b) Um socket UDP IPv6
 - c) Um socket TCP IPv6
 - d) Um socket TCP IPv4
16. [1 valor] Num sistema multicore (e.g., um intel i7), o desativar das interrupções é um método de exclusão mútua porque...
- a) Impede que o processo em execução seja retirado do processador, impossibilitando pois que haja concorrência
 - b) impede que ocorram acesso às caches do processador, garantindo dessa forma a exclusão mútua
 - c) impede que decorram operações de escalonamento por parte do sistema operativo, o que leva a que o mesmo fluxo de execução (*thread* ou processo) se mantenha no processador até que as interrupções sejam novamente reativadas
 - d) falso, pois o desativar das interrupções só garante a exclusão mútua em sistemas single core
17. [1 valor] Considere a variável global `G_Total` e a função `Adiciona` (listagem). Considere que existem duas threads, `TA` e `TB`, que executam simultaneamente as chamadas à função `Adiciona` abaixo mostradas...
- ```
int G_Total = 0;
int Adiciona(int add){
 G_Total += add;
}
```
- `TA → Adiciona(2);`
- `TB → Adiciona(4);`
- Qual é um valor que a variável `G_Total` possa ter no final da execução?
- a) 16
  - b) 0
  - c) 4
  - d) Todas as anteriores

**(Grupo II nas páginas seguintes)**

## Grupo II [3 valores]

Deve escrever com **caligrafia legível**. Respostas ilegíveis **não** são consideradas, resultando na nota 0 (zero valores) à alínea em apreço.

18. [3 valores] Considere a listagem de código C abaixo apresentada compila sem warnings, nem erros.

```
#include "todos.h"
int Command(const char *ipAddress, const unsigned short port){
 struct sockaddr_in RemoteAddr;
 int Sock = -1;
 char Msg_S[128];
 int TotalRecv, Done, Recv, Sent;

 Sock = socket(AF_INET, SOCK_STREAM, 0);
 if (Sock == -1){
 fprintf(stderr, "Can't create socket\n");
 exit(1);
 }
 memset(&RemoteAddr, 0, sizeof(struct sockaddr_in));
 RemoteAddr.sin_family = AF_INET;
 RemoteAddr.sin_port = htons(port);

 inet_pton(AF_INET, ipAddress, (char*)&(RemoteAddr.sin_addr.s_addr));
 if(connect(Sock, (struct sockaddr *)&RemoteAddr, sizeof(RemoteAddr))==-1){
 fprintf(stderr, "Can't connect\n");
 exit(1);
 }

 sprintf(Msg_S, "GET / HTTP/1.0\r\n\r\n");
 Sent = send(Sock, Msg_S, strlen(Msg_S), 0);
 if(Sent == -1){
 fprintf(stderr, "Error on send\n");
 exit(1);
 }
 else{
 printf("[INFO] %d bytes sent\n", Sent);
 }

 /* WHILE */
 TotalRecv = 0;
 Done = 0;
 while(! Done){
 Recv = recv(Sock, Msg_S, sizeof(Msg_S)-1, 0);
 if (Recv == -1){
 fprintf(stderr, "Error on recv\n");
 }
 else if(Recv == 0){
 printf("RECV done!\n");
 break;
 }
 else{
 TotalRecv += Recv;
 Msg_S[Recv]='\0';
 printf("RECV='%s' (Total so far:%d bytes)\n",
 Msg_S, TotalRecv);
 }
 }
 if (Sock != -1){
 close(Sock);
 }
 return 0;
}

int main(void){
 /* IP for www.google.com */
 Command("74.125.230.176", 80);
 return 0;
}
```

**== EXEMPLO DE SAÍDA QUANDO SE EXECUTA A APLICAÇÃO ==**

```
[INFO] 18 bytes sent
RECV='HTTP/1.0 302 Found

Location: http://www.google.pt/
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Set-Cookie' (Total so far:127 bytes)
RECV=': PREF=ID=7bf4bef0d0bdb49d:FF=0:TM=1295826856:LM=1295826856:S=RLjMV6-zmztmyxWY;
expires=Tue, 02-Jan-2020 23:54:16 GMT; path=/; ' (Total so far:254 bytes)
RECV='domain=.google.com
Set-Cookie:
NID=43=LsjCD3JrdoQHqLCAQTmpPwMH00RtqxP9FqtLjiPSTD_i6ph1fqGcNFzrEZ6j1317j6PIytGlyE0qfDlc4zkred
oM' (Total so far:381 bytes)
RECV='Riv0pgiHVb4BbZ9KTYwvuZRw30f8yGMvX4FMd2LO; expires=Mon, 25-Jul-2011 23:54:16 GMT;
path=/; domain=.google.com; HttpOnly
Date: Su' (Total so far:508 bytes)
RECV='n, 03 Jan 2018 23:54:16 GMT
Server: gws
Content-Length: 218
X-XSS-Protection: 1; mode=block

<HTML><HEAD><meta http-equiv=' (Total so far:635 bytes)
RECV='\"content-type\" content=\"text/html; charset=utf-8\">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has mov' (Total so far:762 bytes)
RECV='ed
here.

</BODY></HTML>
' (Total so far:824 bytes)
RECV done!
```

a) Qual é o protocolo de transporte empregue? Justifique convenientemente.

---

---

---

---

b) Qual é o propósito da aplicação? Justifique convenientemente.

---

---

---

---

---

c) A função recv(...) está a ser chamada no interior de um ciclo while. Porquê?

---

---

---

---

---

---

---