

**Punktacja:** Za każde pytanie testowe 2 pkt, za zadanie maksymalnie 20 pkt. 30 punktów gwarantuje zaliczenie.

1. Pamięć systemowa składa się z 32 ramek. Zgodnie ze strategią bliźniaków (ang. buddy), tak jak to robi Linux 2.0.x, przydzieliliśmy 11 ramek. Podaj ile wolnych bloków i o jakim rozmiarze pozostanie w systemie buddy po przydzieleniu. Wynik podaj w postaci: rozmiar bloku (liczba ramek) – liczba takich bloków pozostała w systemie.
2. Wyjątek stronicowania może zostać wygenerowany (A) Przez proces działający w trybie użytkownika próbujący zapisać dane pod adresem uzyskany m w wyniku pomyślnego wywołania funkcji `malloc()` (B): Przez proces działający w trybie jądra, próbujący zapisać dane pod adresem uzyskany m w wyniku pomyślnego wywołania funkcji `vmalloc()`. (C) Obydwie odpowiedzi A i B są prawidłowe (D) Obydwie odpowiedzi A i B są nieprawidłowe.
3. Która operacja NIE jest wykonywana podczas przełączania kontekstu? (A) Zapisanie stron wywłaszczanego procesu na dysk i zwolnienie zajmowanych przez nie ramek. (B) Zapisanie stanu rejestrów wywłaszczanego procesu. (C) Opróżnienie bufora TLB (D) Wszystkie operacje A,B,C są wykonywane.
4. Które z poniższych funkcji/makrodefinicji możemy wywołać wewnątrz handlera obsługi przerwania? (A) `kmalloc(int,GFP_ATOMIC)` (B) `schedule()` (C) `get_user/put_user` (D) nie wolno użyć żadnej z trzech.
5. Jakie fragmenty kodu: (A) kod procesu użytkownika (B) kod dolnej połowy (ang. bottom half)(C) kod handlera obsługi przerwania (D) normalny kod jądra mogą być przerwane przez wykonanie dolnej połowy? Podaj odpowiednie symbole .....
6. Do czego służy funkcja `printk`?
7. Wymień dwie sytuacje powodujące zbudzenie procesu uśpionego poprzez wywołanie funkcji `interruptible_sleep_on`.
8. Kolejka żądań urządzenia blokowego jest pusta. Po wstawieniu żądania do kolejki jądro wywoła ..... sterownika urządzenia.
9. Funkcja `bread` jest odpowiedzialna za odczyt bufora z urządzenia blokowego. Wywołanie tej funkcji wiąże się **zawsze** z (A) przeszukaniem tablic mieszających w pamięci buforowej w celu odnalezienia bufora. (B) fizycznym odczytem bufora z dysku (C) alokacją pamięci dla nowego bufora (D) wszystkie odpowiedzi A-C są błędne.
10. Różnica pomiędzy funkcjami zwalnającymi bufor `brelease` oraz `bforget` polega na tym że:
11. Podaj przynajmniej jedną przyczynę, dla której warstwa VFS systemu Linux używa oddzielnych obiektów dla i-węzła oraz otwartego pliku.
12. W jakiej sytuacji jądro Linuksa nie pozwoli na usunięcie modułu np. poleceniem `rmmod` (Odpowiedź nie więcej niż jedno zdanie)

13. Kod programu pewnego zajmuje dokładnie 32Kbajty. Program nie korzysta z bibliotek współdzielonych. Użytkownik uruchomił 4 kopie tego programu. Po uruchomieniu 30 kopii kod wykorzystywany przez wszystkie kopie zajmuje w pamięci RAM komputera (A) 128 Kbajty (B) 32Kbajty (C) żadna z poprzednich odpowiedzi nie jest prawidłowa.

14. W przypadku typowej karty sieciowej kopiowanie odebranego pakietu z urządzenia sieciowego do pamięci komputera następuje następuje (A) na żądanie jądra wyrażone poprzez wywołanie funkcji read sterownika. (B) wewnątrz handlera obsługi przerwania karty będącego częścią sterownika. (C) w cyklicznie powtarzających się odstępach (D) wszystkie poprzednie odpowiedzi są błędne.

15. W systemie Linux 2.0.x proces potencjalne punkty wyłączenia procesu to: (A) powrót z szybkiego przerwania (B) powrót z wolnego przerwania (C) powrót z wywołania systemowego. Poprawne odpowiedzi to .....

16. W Uniksowych systemach plików (np. ext2 Linuksa) atrybuty pliku (np. długość, czasy utworzenia, dostępu i modyfikacji) przechowywane są w (A) i-węźle pliku (B) katalogu (C) super-bloku (D) Liście ACL pliku.

17. Jakie dokładnie warunki musi spełniać proces X, aby być w stanie wyłączyć proces z klasą czasu rzeczywistego SCHED\_FIFO ?

18. Urządzenie wejścia-wyjścia wykorzystuje kontroler DMA. Bufor ma długość 16KB. Jakiej funkcji jądra **\*\*nie\*\*** wolno użyć do przydzielenia pamięci dla bufora tego urządzenia: (A) vmalloc (B) kmalloc (C) \_\_get\_free\_pages (D) wolno użyć każdej z tych trzech funkcji.

19. Jeżeli zależy nam na najszybszym odczycie danych z pliku na **\*\*bardzo\*\*** szybkim dysku (np. macierz RAID) to należy użyć funkcji: (A) mmap (B) read (C) Nie ma żadnej różnicy pomiędzy szybkością w przypadkach (A) oraz (B).

20. Sterownik, z powodu błędu programisty, w metodzie read usiłuje odwołać się do nie istniejącego (niepoprawnego) adresu jądra. Spowoduje to (A) panikę jądra (B) komunikat "Oops" i przerwanie procesu, który wywołał funkcję read (C) zmodyfikowanie przypadkowych danych pod tym adresem, co w konsekwencji może prowadzić do zakłócenia pracy jądra (D) wszystkie odpowiedzi A-C są błędne.

**Zadanie:** Dane jest urządzenie znakowe, które działa w sposób następujący: odczyt z portu o numerze  $p$  równy 0 oznacza, że urządzenie gotowe jest do przyjęcia następnego bajtu, Odczytana wartość różna od 0 oznacza, że jeszcze nie można wysłać bajtu. Wysłanie bajtu następuje poprzez jego zapis do portu  $p$ . Napisz sterownik urządzenia, funkcjonujący jak moduł jądra i obsługujący cztery urządzenia o numerach podrzędnych 0,1,2,3 i korzystające odpowiednio z portów 0x200,0x201,0x202,0x203. Zakładamy, że indywidualne urządzenie obsługuje wyłącznie zapis i w danej chwili może być otwarte co najwyżej przez jeden proces, Kolejne procesy usiłujące otworzyć urządzenie powinny być wstrzymywane do momentu jego zamknięcia przez proces, który je aktualnie otworzył. Podaj kod funkcji open, release, oraz write. Implementacja load\_module i cleanup\_module nie jest potrzebna. Należy uwzględnić fakt, że urządzenie jest \*\*\*wyjątkowo\*\*\* wolne i operacja write może potrwać kilkadziesiąt sekund. W związku z tym mile widziane jest zapewnienie że proces wywołujący write nie zmonopolizuje procesora na czas kilkadziesiątu sekund, oraz że proces będzie reagował na sygnały.

Zmienne jądra:

need\_resched – aktualny proces zużył cały kwant czasu.

current->signal – maska bitowa zgłoszonych sygnałów.

current->blocked – maska bitowa zablokowanych sygnałów.

Kody błędów przy powrocie z funkcji:

-ERESTARTSYS – proces przerwany przez sygnał.

-ENODEV – nie ma takiego urządzenia.

Funkcje i makrodefinicje jądra:

```
void schedule();
```

```
void *kmalloc(size,priority); // priority GFP_KERNEL albo GFP_ATOMIC
```

```
void kfree(void *ptr);
```

```
void down(struct semaphore *sem); // wejście
```

```
void up(struct semaphore *sem); // wyjście
```

```
MOD_INC_USE_COUNT;
```

```
MOD_DEC_USE_COUNT;
```

```
char inb(short port); // odczyt bajtu z portu
```

```
void outb(short port,char byte); // zapis bajtu do portu
```

```
char get_user(char *user_address);
```

Funkcje sterownika, które trzeba zaimplementować:

```
int open(struct inode *inode, struct file *file);
```

```
int release(struct inode *inode, struct file *file);
```

```
int write(struct inode *inode, struct file *file,char *buffer,int size);
```

Numer podrzędny urządzenia: MINOR(inode->i\_rdev);