

## Test pitanja za vežbu u vezi bloka tema 7

1. Zaokruži tačne tvrdnje u vezi super-petlje:

- to je petlja koja se beskonačno ponavlja
- super-petlja služi da se ne bi morali raditi višestruki pozivi potprograma
- u okviru super-petlje se obično realizuje kompletan hardverski zadatak
- osnovna osobina super-petlje je paralelno izvršavanje pojedinih programskih blokova

2. Označi osobine event-driven (na bazi događaja) programa:

- nepotrebno blokiranje rada zbog pauza se eliminiše
- dodavanje novih funkcija se lako izvodi
- indikator događaja može imati proizvoljno mnogo stanja
- indikator događaja se koristi isključivo u okviru prekidnih rutina

3. Za softverske FIFO bafere važi:

- FIFO baferi se koriste najčešće kod direktne obrade prekida
- kod FIFO bafera podatak koji prvi uđe u bafer poslednji izlazi
- FIFO baferi se koriste u slučajevima kada imamo ponavljane događaje koji zahtevaju složeniju obradu, a pojavljuju se u unapred nepoznatom trenutku
- jedan od problema koje FIFO bafer može da reši je i problem gubitka podataka koji učestalo dolaze (npr. preko serijskog porta)

4. Označi minimalno neophodne elemente za realizaciju softverskog FIFO bafera pomoću

A) indeksa:      B) pokazivača:

- konstanta koja predstavlja veličinu FIFO bafera
- dva pokazivača koji služe za praćenje upisa/čitanja
- dva indeksa koji služe za praćenje upisa/čitanja
- brojač elemenata FIFO-a
- indikator koji označava da je FIFO prazan
- indikator koji označava da je FIFO pun
- indikator da je došlo do preklapanja pokazivača za čitanje i upis

5. Označi indikatore događaja u okviru sledećeg kodnog primera:

```
...
```

<pre>void merenje(void) {     if (isr_100)     {         isr_100 = 0;         .... // merenje         if (brojac==10)         {             brojac = 0;             prikaz_OK = 1;         }     } }</pre>	<pre>void obrada_tastature(void) {     if (kbhit())     {         ....// rad sa         tastaturom     } } void prikaz(void) {     if (prikaz_OK)     {         prikaz OK = 0;</pre>	<pre>void isr_timer(void) interrupt... { /* interval == 1ms */     if (++timer==100)     {         timer = 0;         isr_100 = 1;     } }</pre>
--	--	--

	<pre> .....// prikaz rezultata     } } </pre>	
--	---	--

-prikaz\_OK  
 -brojac  
 -kbhit()  
 -timer  
 -isr\_100

6. Zaokružiti netačan/netačne iskaze vezane za superpetlju :

- Za ovu metodu se često koristi naziv Round Robin metoda.
- Superpetlja je petlja u kojoj se realizuje kompletan hardverski zadatak.
- Osnovna karakteristika ove metode je sekvencijalno izvršavanje pojedinih programskih blokova, uključivši i eventualna kašnjenja (pauze).
- Glavna prednost je pauza.

7. Poveži različite implementacije FIFO bafera sa njihovim karakteristikama:

- Implementacija pomoću ulančanih lista
- Implementacija pomoću indeksa
- Implementacija pomoću pokazivača

- Pravilnim odabiranjem veličine bafera se može postići dovoljno efikasan kod što je dobro za realizaciju na slabijim mikrokontrolerima
- Za svaki element se dinamički alocira memorija pri upisu, a pri čitanju se oslobadja
- Maksimalan broj elemenata je za jedan manji od ukupne veličine bafera

8. Šta je event (izabrati tačne odgovore)?

- Hardverska indikacija hardverske promene
- Softverska indikacija hardverske promene
- Hardverska indikacija softverske promene
- Softverska indikacija softverske promene
- nastup grupe Queen na Live AID koncertu 1985 😊

9. Označiti tačne iskaze vezane za FIFO bafere fiksne veličine sa pokazivačima:

- kada je bafer prazan pokazivači nisu izjednačeni
- u bafer se može smestiti najviše za jedan element manje od njegove veličine
- koristi se više od dva pokazivača
- broj elemenata u baferu se određuje razlikom pokazivača upisa i čitanja

10. Označiti netačne iskaze vezane za pojam događaja - Event:

-Event predstavlja uvođenje pojma događaja, koji predstavlja hardversku indikaciju softverske ili hardverske promjene.

-Indikator događaja ima samo jedno moguće stanje i obično je veličine jednog bita, ili jednog bajta.

-Event Driven predstavlja kontrolu izvršavanja pomoću događaja.

-Ako se događaj testira samo radi aktiviranja jedne funkcije, tada je ovaj test moguće sprovesti unutar te funkcije, a ne u glavnoj petlji.

11. Zaokruži tačnu tvrdnju/tvrdnje koje se odnose na FIFO bafer:

a) Naziva se još i cirkularni bafer

b) Podatak koji je poslednji ušao u bafer prvi izlazi iz njega

c) Koristi se kada je potrebno sačuvati informacije koje ne mogu odmah da se obrade

d) Primitljeni podaci se obraduju nasumično

12. Označi tačno tvrdjenje/tvrdjenja za sledeću super-petlju:

(indikator isr\_50 se aktivira na svakih 50ms u prekidu tajmera)

```
void main(void)
{
    int i = 0;
    while (1)
    {
        if (kbhit())
            obrada_tastature();

        if (isr_50)
        {
            isr_50 = 0;
            izvrsiMerenje();
            if ((++i)==10)
            {
                i = 0;
                prikaz_merenja = 1;
            }
        }

        if (prikaz_merenja)
        {
            prikaz_merenja = 0;
            prikaziMerenja();
        }
    }
}
```

a) ova petlja sadži 3 indikatora događaja

b) prednost ove realizacije je što se ni u jednom trenutku ne blokira izvršavanje programa

c) funkcija izvrsiMerenje() će se pozvati četvrti put nakon 0.2 sekunde

d) funkcija prikaziMerenja() se poziva nakon 9 merenja

13. Označi tačne tvrdnje vezane za FIFO bafer sa ulančanim nizovima:

- a) Pri upisu u bafer koristi se dinamička alokacija memorije
- b) Pri čitanju iz bafera element se ne briše iz memorije
- c) Metoda se često koristi zbog svoje jednostavnosti
- d) Slično FIFO baferu sa pokazivačima, sadrži pokazivač na naredni element

14. Označi rezultat

```
#include <stdio.h>
#define MAX_SIZE 3

unsigned char rd_index = 0, wr_index = 0, length = 0;
char buffer[MAX_SIZE];

void fifo_write(char data) {
    if (length < MAX_SIZE) {
        buffer[wr_index++] = data;
        if (wr_index >= MAX_SIZE)
            wr_index = 0;
        length++;
    }
}

char fifo_read(void) {
    if (length) {
        char tmp = buffer[rd_index++];
        if (rd_index >= MAX_SIZE)
            rd_index = 0;
        length--;
        return tmp;
    }
    return 0;
}

int main(){
    unsigned char data1, data2;

    fifo_write(10);
    fifo_write(12);
    fifo_write(14);
    fifo_write(16);
    data1 = fifo_read();
    data2 = fifo_read();
    fifo_write(18);
    printf("%d, ", wr_index);
    printf("%d, ", rd_index);
    printf("%d\n", length);
    return 0;
}
```

- A) 0, 2, 2      B) 1, 2, 2      C) 5, 2, 3      D) 5, 2, 5

15. Pridružiti programima pod a i b osobine označene od 1 do 4:

```
void main(void)
{
    while (1)
    {
        obrada_tastature();
        if (do_10kg)
            merenje1();
        else
            merenje2();
        prikaz();
    }
}
```

```
#define do_10kg 0
void main(void)
{
    char i = do_10kg;
    while (1)
    {
        if (i==do_10kg)
            i = merenje_do_10kg();
        else
            i = merenje_preko_10kg();
    }
}
```

a) selekciona

b) radna

- 1) f-je za obradu tastature i prikaz rezultata su zajedničke
- 2) program sastavljen od glavne i 2 radne petlje
- 3) Na osnovu obrade tastature može se prekinuti izvršenje petlje
- 4) Funkcije merenja sadrže super petlju

16. Na osnovu date strukture promenljivih unutar realizacije FIFO buffera, odrediti o kojem od poznatih tipova realizacije je reč (uz pretpostavku da se sve promenljive koriste):

```
typedef struct fifo_buffer {
    void *podaci;
    size_t kapacitetBuffera;
    size_t brojClanova;
    void *head;
    void *nextW
    void *nextR
} fifo_buffer;
```

- a) FIFO sa ulančanim nizovima
- b) FIFO fiksne veličine sa indeksima
- c) FIFO fiksne veličine sa pokazivačima
- d) Nije moguće odrediti bez funkcionalne implementacije