Zadatak 4 – Decembar 2020

(*8 poena*) Definisati 8086 kod i izgled aktivacionog sloga za funkciju *allocate*. Za predstavljanje podataka tipa int kao i memorijskih adresa koriste se 2 bajta. Pretpostaviti da se rezultat funkcije *allocate* smešta na stek.

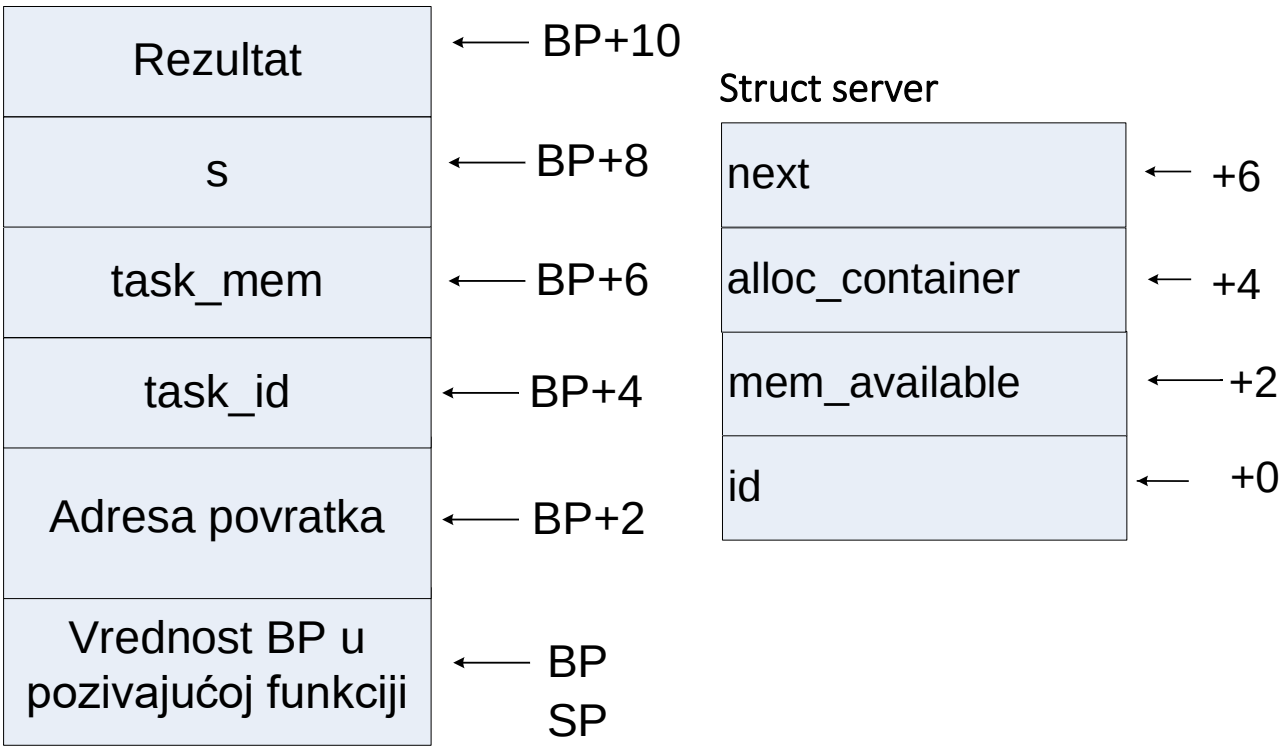
```
struct server{
  int id;
  int mem_available;
  int alloc_container;
  struct server* next;
};
```

```
int allocate(int task_id, int task_mem, struct
server* s){
  if (task_mem < s->mem_available){
    s->mem_available = s->mem_available-task_mem;
    s->alloc_container=task_id;
    return s->id;
  }
  if(s->next!=0)
    return allocate(task_id,task_mem,s->next);
  else return 0;
}
```

Komentar: Sve promenljive int tretiramo da su 16bit, a podrazumevamo da se kontekst procesora čuva na početku, a restaurira na kraju potprograma. Zbog toga su ovi delovi u rešenju izostavljeni. Osim toga, interesuje nas samo 16bit rezultat množenja iz AX, po dogovoru.

Napomena: Moguće je da u rešenju postoje greške, tako da su sugestije na mail dobrodošle. Osim toga, pojedine optimizacije prilikom generisanja koda u ovom rešenju nisu primenjene.

Aktivacioni slog



8086 kod:

upis BP na stek

```
PUSH	BP
```

promena vrednosti pokazivaca BP:

```
MOV	BP, SP
```

Telo funkcije:

Pribavi *s:

```
MOV	BX, [BP+8]
```

Pribavi s->mem_available u AX:

```
MOV	AX, [BX+2]
```

Pribavi task_mem u DX:

```
MOV	DX, [BP+6]
```

Uporedi:

```
CMP	DX, AX
JGE	ELSE_LABELA
THEN_LABELA:
SUB AX, DX
MOV [BX+2], AX
MOV CX, [BP+4]
MOV [BX+4], CX
MOV AX, [BP]
```

Vrati rezultat:

```
MOV [BP+10],AX
JMP KRAJ
ELSE_LABELA:
MOV	AX, [BX+6]
CMP	AX, 0
JE	ELSE_LABELA2
```

Alociranje mesta za rezultat na steku:

```
PUSH AX
```

Prvi argument:

```
MOV AX, [BX+6]
PUSH AX
```

Drugi argument:

```
MOV AX, [BP+6]
PUSH AX
```

Treći argument:

```
MOV AX, [BP+4]
PUSH AX
```

Poziv fukcije:

```
CALL ALLOCATE
```

Skidanje parametara sa steka:

```
ADD SP, 6
```

Pribavi rezultat sa steka:

```
POP AX
```

Vrati rezultat kroz return:

```
MOV [BP+10], AX
JMP KRAJ
ELSE_LABELA2:
MOV	[BP+10], 0
KRAJ:
```

Vraćanje SP na vrednost BP:

```
        MOV         SP,BP
```
Uzimanje stare vrednosti BP-a:
```
        POP         BP
        RET
```