

## Zadatak 4 – Januar 2020

(8 poena) Definirati 8086 kod i izgled aktivacionog sloga za funkciju *render\_scene*. Za predstavljanje podataka tipa *int* kao i memorijskih adresa koriste se 2 bajta. Pretpostaviti da se rezultat funkcije *load\_model* smešta na stek, a rezultat funkcije *render\_scene* u registar CX.

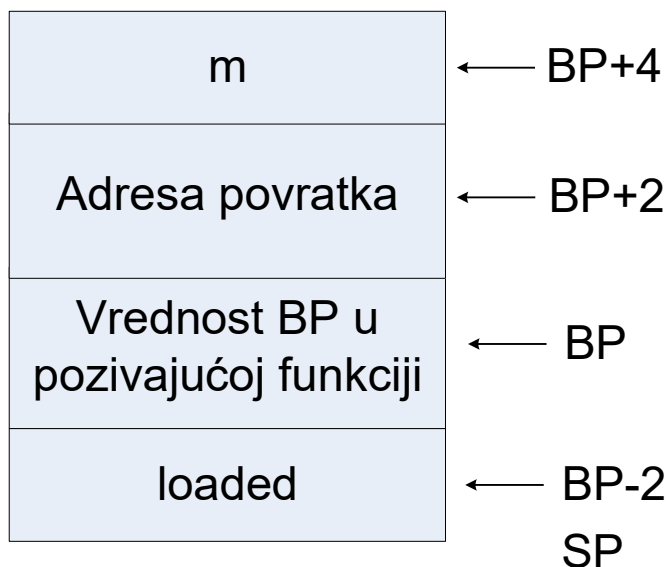
```
struct model {
    int id;
    int type;
    struct model *next;
};

int render_scene(struct model* m) {
    int loaded = 0;
    loaded = loaded + load_model(m->id, m->type);
    if(m->next != 0)
        loaded += render_scene(m->next);
    return loaded;
}
```

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



### Struct model



## 8086 kod:

upis BP na stek

```
PUSH    BP
```

promena vrednosti pokazivaca BP:

```
MOV     BP, SP
```

Rezervacija prostora za lokalnu promenljivu loaded, veličine 2 byte int:

```
SUB     SP, 2
```

Telo funkcije:

```
MOV     [BP-2], 0
```

Pribavi \*m

```
MOV     BX, [BP+4]
```

Poziv load\_model:

```
PUSH    AX
```

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

```
PUSH    AX
```

```
MOV     AX, [BX]
```

```
PUSH    AX
```

```
CALL    LOAD_MODEL
```

```
ADD     SP, 4
```

```
POP     AX
```

Sabiranje rezultata sa prethodnim loaded:

```
MOV     DX, [BP-2]
```

```
ADD     DX, AX
```

```
MOV     [BP-2], DX
```

Provera m->next:

```
MOV     AX, [BX+4]
```

```
CMP     AX, 0
```

```
JE      KRAJ
```

Rekurzivni poziv:

```
MOV     AX, [BX+4]
```

```
PUSH    AX
```

```
CALL    RENDER_SCENE
```

```
ADD     SP, 2
```

```
MOV     DX, [BP-2]
```

```
ADD     DX, CX
```

```
MOV     [BP-2], DX
```

KRAJ:MOV CX, [BP-2]

Izbacivanje lokalnih promenljivih iz steka:

```
MOV     SP, BP
```

Uzimanje stare vrednosti BP-a:

```
POP     BP
```

```
RET
```