

Zadatak 4 – Septembar 2020

1. (8 poena) Definirati 8086 kod i izgled aktivacionog sloga za funkciju *covid_safety*. Za predstavljanje podataka tipa *int* kao i memorijskih adresa koriste se 2 bajta. Pretpostaviti da se rezultat funkcije *call_guard* (koji je tipa *int*) smešta na stek, a rezultat funkcije *covid_safety* vraća kroz registar CX.

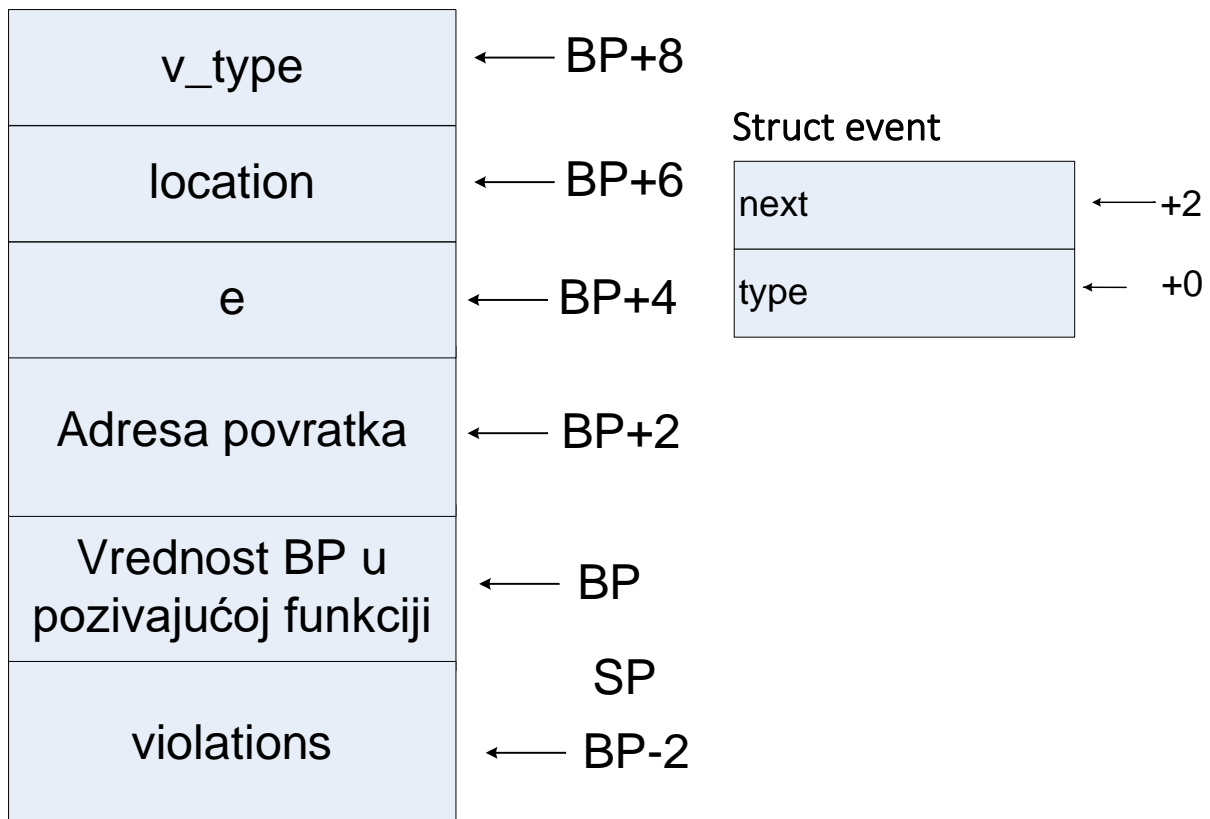
```
struct event{
    int type;
    struct event* next;
};

int covid_safety(struct event *e, int location, int v_type) {
    int violations = 0;
    if (e->type == v_type) {
        violations += call_guard(v_type, location);
    }
    if(e->next != 0)
        return violations + covid_safety(e->next, location, v_type);
}
```

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
```

Rezervacija prostora za lokalnu promenljivu:

```
SUB SP, 2
```

Telo funkcije:

Pribavi *e:

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

Pribavi e->type u AX:

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

Pribavi v_type u DX:

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

Uporedi:

```
CMP DX, AX
```

```
JNE ELSE_LABELA
```

THEN_LABELA:

Pribavi location u CX:

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

Alociranje mesta za rezultat na steku:

```
PUSH AX
```

Prvi argument:

```
PUSH CX
```

Drugi argument:

```
PUSH DX
```

Poziv fukcije:

```
CALL CALL_GUARD
```

Skidanje parametara sa steka:

```
ADD SP, 4
```

Pribavi rezultat sa steka:

```
POP AX
```

```
ADD [BP-2], AX
```

ELSE_LABELA:

Proveravamo e->next da li je različigo od 0

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

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

```
CMP AX, 0
```

```
JE KRAJ
```

Prvi argument:

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

```
PUSH AX
```

Drugi argument:

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

```
PUSH AX
```

Treći argument:

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

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

```
PUSH AX
```

Poziv fukcije:

```
CALL COVID_SAFETY
```

Skidanje parametara sa steka:

```
ADD SP, 4  
ADD [BP-2], CX
```

Vraćanje rezultata kroz CX:

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

KRAJ:

Vraćanje SP na vrednost BP:

```
MOV SP, BP
```

Uzimanje stare vrednosti BP-a:

```
POP BP
```

```
RET
```