

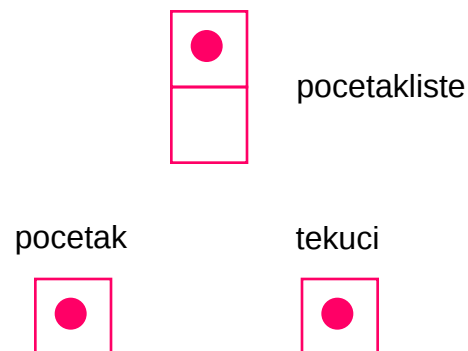
Povezane strukture podataka

(Poludinamičke i dinamičke strukture podataka)

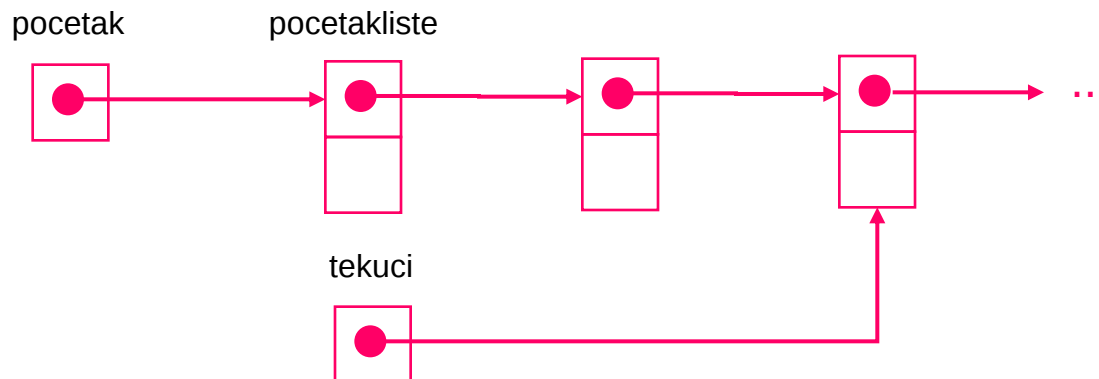
Jednostruko povezane liste

- skup čvorova povezanih pokazivačima u jednom smeru.
- svaki čvor je strukturna promenljiva
- lista može da sadrži promenljiv broj čvorova

```
typedef struct cvor_st {  
    struct cvor_st *sledeci;  
    /* ... */  
} JCVOR;  
  
JCVOR pocetakliste, *pocetak, *tekuci;
```



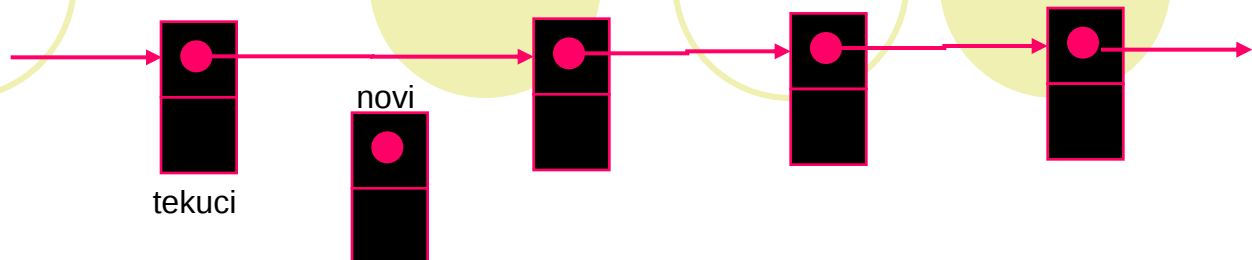
OPŠTI OBLIK JEDNOSTRUKO POVEZANE LISTE



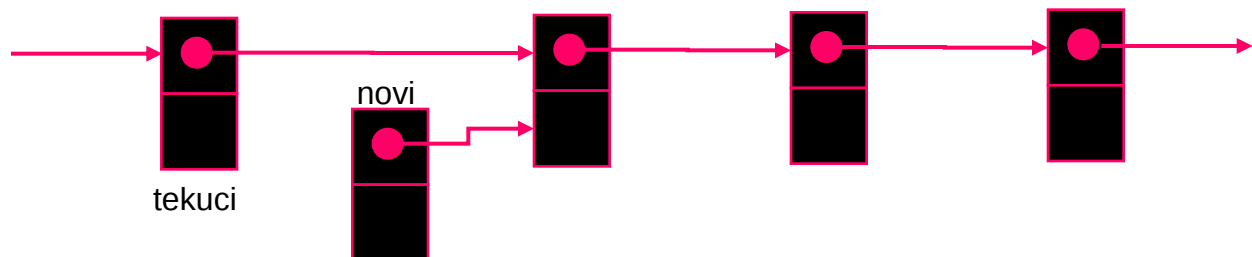
Operacije nad listom

- umetanje čvora u povezanu listu
- brisanje tekućeg čvora iz liste
- pristup čvoru radi čitanja i upisa

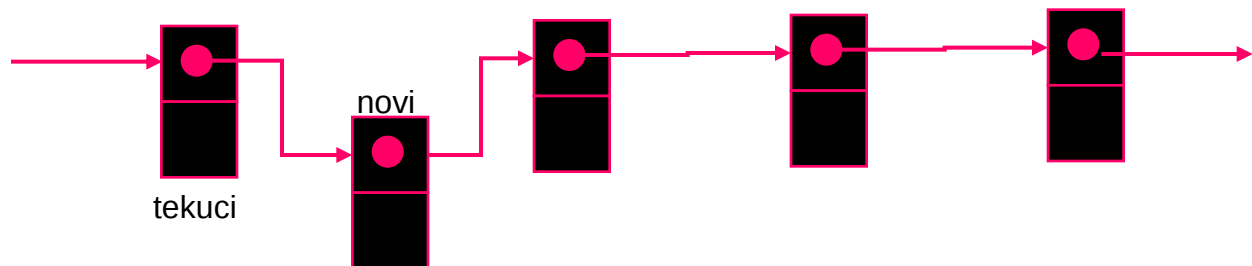
```
novi = (JCVOR *)malloc(sizeof(JCVOR));
```

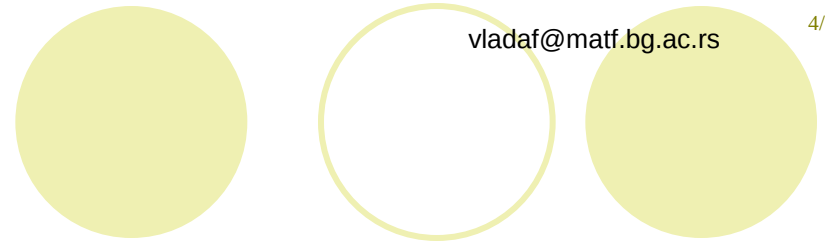


```
novi->sledeci = tekuci->sledeci;
```



```
tekuci->sledeci = novi;
```





```
JCVOR *  
umetni_cvor(JCVOR *listp)  
{  
    JCVOR *novi;  
  
    novi = (JCVOR *)malloc(sizeof(JCVOR));  
    if (novi != NULL); {  
        novi->sledeci = listp->sledeci;  
        listp->sledeci = novi;  
    }  
    return novi;  
}
```

```
void  
brisi_cvor(JCVOR *listp)  
{  
    JCVOR *priv;  
  
    priv = listp->sledeci;  
    listp->sledeci = listp->sledeci->sledeci;  
    free (priv);  
}
```

Ciklične jednostruko povezane liste

pocetak->sledeci = pocetak;

Upravljanje memorijom

```
/*
 * strdup.c
 *
 * Napravi bafer odgovarajuće veličine i kopiraj u njega znakovni niz.
 */

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

char *
strdup(const char *niz)
{
    char *novi;

    novi = (char *)malloc(strlen(niz) + 1);
    return novi == NULL ? novi : strcpy(novi, niz);
}

int
main ()
{
    char niz [] = "Probni niz!";
    char *novi;

    printf ("\nPocetni niz : %s", niz);
    novi = strdup (niz);
    printf ("\nDuplirani niz: %s", novi);

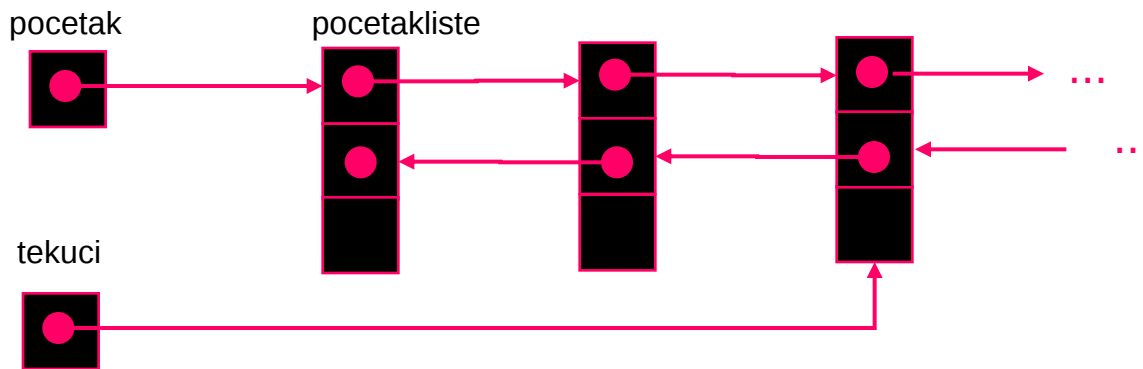
    return EXIT_SUCCESS;
}
```

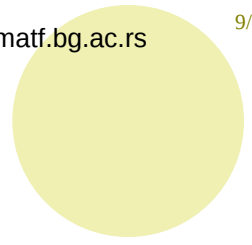
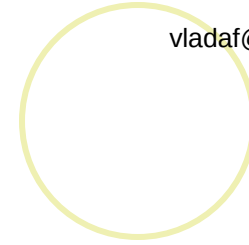
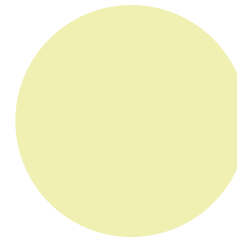
Primer: kartoteka

Dvostruko povezane liste

```
typedef struct dcvor_st {  
    struct dcvor_st *sledeci, *prethodni;  
    /* ... */  
} DCVOR;  
  
DCVOR pocetakliste, *pocetak, *tekuci;
```

OPŠTI OBLIK DVOSTRUKO POVEZANE LISTE





```
DCVOR *
umetni_dcvor(DCVOR *listp)
{
    DCVOR *novi;

    novi = (DCVOR *)malloc(sizeof (DCVOR));
    if (novi != NULL); {
        novi->prethodni = listp;
        novi->sledeci = listp->sledeci;
        listp->sledeci ->prethodni = novi;
        listp->sledeci = novi;
    }
    return novi;
}
```

```
void
brisi_cvor(DCVOR *listp)
{
    listp->prethodni->sledeci = listp->sledeci;
    listp->sledeci->prethodni = listp->prethodni;
    free (listp);
}
```

Ciklične dvostruko povezane liste

CIKLIČNA LISTA

