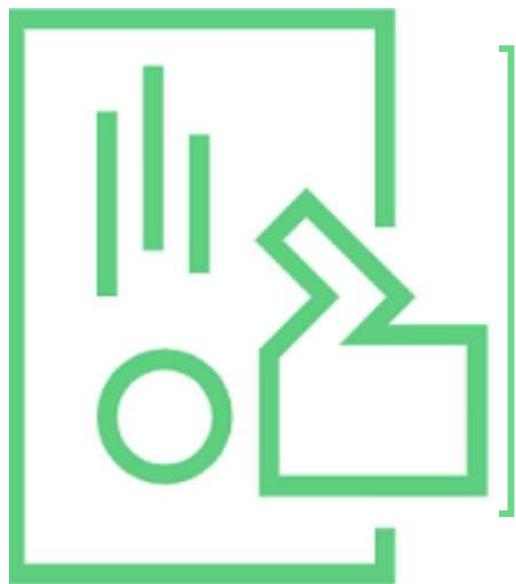


Dynamiska Arrayer

Stefan Holmberg, Systementor AB

1



Dynamiska arrayer

Behövs när du inte vet hur stor listan kommer att vara
T.ex när du inte vet hur många SPELARE eller HUNDAR ditt "register" ska inehålla

Deklareras som antingen en array av okänd storlek:

```
// inte för oss !!! och vår Warning level  
// int lista[];
```

Eller som en pekare:

```
int *lista;
```

De är ekvivalenta sedan

Dynamisk array: deklaration

```
int main() {  
    int *a;  
  
    return 0;  
}
```

Deklarera en dynamisk lista
med en pekare (använd
malloc)

Tänk att en pekare pekar ut första positionen. Du kan
ALLTID göra pekare[10] osv som flyttar fram värdet
10 positioner

Dynamisk array: malloc

Men när du deklarerar en dynamisk lista så fördelas endast minne för en pekare till en lista: listan inte ännu

Vi måste be datorn om minne till vårt array med *malloc()*

malloc() deklaras i *stdlib.h*:
`void* malloc(size_t size);`

malloc():

Tar som parameter hur mycket minne du behöver (i bytes)

Fördelar minne

Returnerar adressen till minnet

Dynamisk Array: malloc

Hur vet vi hur mycket minne vi behöver?

Använd `sizeof()`

`sizeof(int) = 4`

`sizeof(float) = 4`

`sizeof(char) = 1`

`sizeof(char*) = 4` - ?!

Dynamisk Array: malloc

```
int foo( int n ) {  
    int *a = malloc( sizeof(int) * n );  
    a[1] = 10;  
    free(a);  
    return 0;  
}
```

Fördela minne :

- n = storlek på listan
- sizeof(int) = storlek på varje element i listan
- malloc(size) = fördela minne
- Listan kan nu användas

Dynamisk Array: malloc

Kan minnet ta slut? Ja!

malloc() returnerar 0 (= NULL, en speciell pekare)

OBS: man kan inte lita på att malloc() alltid lyckas fördela minne
Sällsynt men det kan hända
Mer vanligt i IoT miljöer än på PC

Vår för tar minnet slut?

Program som löpande fördelar minne som de sedan inte ger tillbaka när det inte behövs
längre
Använd *free()*!

Dynamisk Array: free

free() deklaras också i *stdlib.h*:

```
void free(void *);  
free( ptr );
```

Tar som parameter en pekare till minne som du har tidigare fördelat med `malloc`
Ger tillbaka minnet till datorn
Returnerar ingenting

OBS: din pekare pekar fortfarande till adressen för minnet, men minnet är
inte fördelat till dig längre!

Dynamisk Array: free

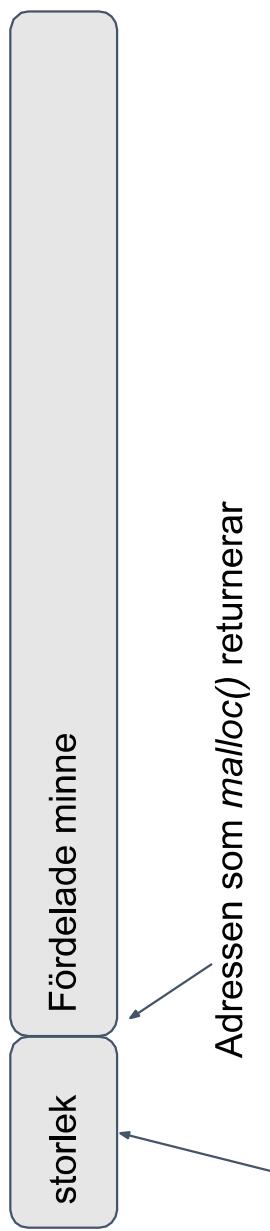
```
int main() {  
    int *ptr = (int *)  
        malloc(sizeof(int));  
    /* Gör något */  
    free(ptr);  
    return 0;  
}
```

Tumregel

- Om du anropar malloc()
- Så ansvarar du för att anropa free()

Dynamisk Array: free

- Hur vet `free()` hur mycket minne var fördelat och ska nu ges tillbaka till datorn?
- `malloc()` fördelar lite mer minne än du ber om, och lagrar hur mycket minne fördelades i den extra platsen:



4 bytes till storlek

Dynamisk Array: storlek

Hur vet jag hur stor en dynamisk lista är?

Det måste du hålla koll på själv!

Och eventuellt skicka med som parameter till funktioner

Hemläxa:

<http://www.geeksforgeeks.org/using-sizof-operator-with-array-parameters/>

Dynamisk Array: realloc

https://www.tutorialspoint.com/c_standard_library/c_function_realloc.htm