

Computer Arkitektur og Operativ Systemer

Denne forelæsning optages og gøres efterfølgende tilgængelig
på Moodle

MEDDEL VENLIGST UNDERVISEREN, HVIS DU IKKE ØNSKER, AT
OPTAGELSE FINDER STED

This lecture will be recorded and afterwards be made available
on Moodle

PLEASE INFORM THE LECTURER IF YOU DO NOT WANT
RECORDING TO TAKE PLACE

Computer Arkitektur og Operativ Systemer Assembler 2

Forelæsning 4
Brian Nielsen

*Credits to
Randy Bryant & Dave O'Hallaron (CMU)*

Idag

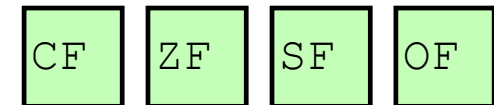
- Hvordan laves kontrol-strukturer i assembler ?
 - `if () { then; } { else; }`
 - `while () { }`
- Hvordan repræsenteres og tilgås sammensatte datatyper
 - Arrays
 - Structs ?

Hvordan evalueres betingelser?

- Hvordan laves højniveau kontrol-strukturer i assembler ?
 - `if (b) { then; } { else; }`
 - `while (b) { }`

1. Evaluering af betingelser: Condition Code Register

Betingelsesflag
Condition codes



Carry (Mente)-flag
Zero (Nul)-flag
Sign (Fortegns)-flag
Overflow (Overløbs)-flag

Hvordan laves betinget forgrening

2. Hops

- Ubetingede: **jmp** <adresse>
- Betingede hop: **JNE** <adresse>

`%rip`

Program tæller
Instruction pointer
(program counter)

3. Oversættelses-skabeloner (med "goto" kode)

- Som mellemtrin fra c-kode til assembler
- Genkende en anvendt skabelon ved "dekompilering" fra ASM til C

C kode

```
long pcount_do
(unsigned long x) {
    long result = 0;
    do {
        result += x & 0x1;
        x >>= 1;
    } while (x);
    return result;
}
```

Goto Version

```
long pcount_goto
(unsigned long x) {
    long result = 0;
loop:
    result += x & 0x1;
    x >>= 1;
    if(x) goto loop;
    return result;
}
```

ASM

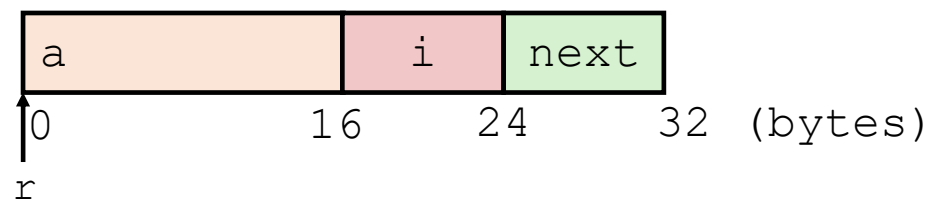
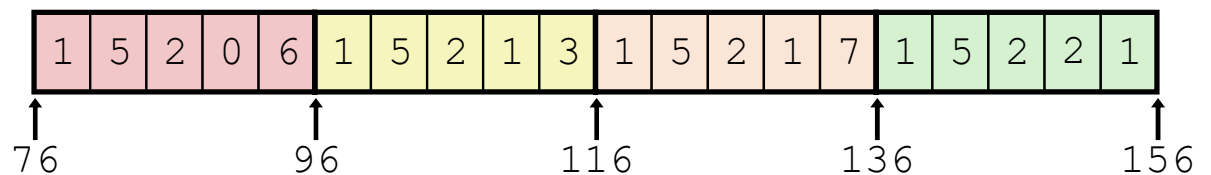
```
##%rdi=x
##%rax=result
movl    $0, %eax    # result = 0
.L2:                                # loop:
movq    %rdi, %rdx   # t= x
andl    $1,%edx      # t= x & 0x1
addq    %rdx, %rax   # result += t
shrq    %rdi         # x >>= 1
jne     .L2          # if (x) goto loop
ret
```

Hvordan layouts sammensatte datatyper?

- Arrays og structs findes ikke på maskine-niveau
- Hvordan udlægges de som stykker hukommelse (i ram)?
 - Arrays i flere dimensioner? → Rækkevist!
 - Heterogene datatyper? → Erklærings-rækkefølge + alignment
- Hvordan indekseres de? → Adresse-beregning

```
int pgh[4][5]=  
  {{1, 5, 2, 0, 6},  
   {1, 5, 2, 1, 3 },  
   {1, 5, 2, 1, 7 },  
   {1, 5, 2, 2, 1 }};
```

```
struct rec {  
  int a[4];  
  size_t i;  
  struct rec *next;  
};
```



Øvelserne

- Fra ASM til goto-kode (de-kompilering af if)
 - Oversættelsestrategi for if
 - Arrays layout & indexing
-
- Challenge 3: Injicér objekt-kode for at forhindre atomkrig
 - Challenge 4: ASM indlejret i C!!



War Games '83

|Hjælpelærer|=Max - Brug dem!

