# Obligatorisk innlevering 2

## Sondre Aasarmoen

## Sivert Sætre

# TASK 1

1)

I linje 5-13: Disse kodene gir oss ord som vi kan bruke senere. Hvis vi for eksempel ønsker å avslutte programmet, et «System Exit», må vi legge inn taller 1 i eax-registeret og kallet på «int 80h». For å gjøre dette enklere for en person som skriver koden setter vi 1 inn i en konstant kalt «SYS\_EXIT», så når vi ønsker å utføre en «System Exit», skriver vi bare order, som da plasserer taller 1 i registeret. I koden ser det ut som «mov eax, SYS\_EXIT» og «int 80h» istedenfor «mox eax, 1» og «int 80h». Det gjør ingen forskjell for programmet, men det gjør det litt lettere å lese og forstå. Med det sagt, så bør vi også skrive «mov ebx, 0 (eller1)» før «int 80h», slik at programavslutningen indikerer en vellykket (0) eller mislykket (1) terminering.

2)

Linje 17 er en del av .bss-seksjonen. Resb i 'siffer resb 4' står for 'reserve bytes' og betyr at vi har laget en variabel 'siffer' som har reservert 4 bytes for senere bruk.

.bss står for 'Block Started by Symbol' og er en seksjon som brukes for å deklarere variabler. Systemet initialiserer vanligvis disse variablene med verdien null. Dette betyr at selv om variablene i denne seksjonen ikke er initialisert av programmereren, pleier operativsystemet typisk å sette disse seksjonene til null og blir kvitt 'søppelverdier'.

3)

Utskriftene skjer i starten av 'global \_start'-seksjonen. Mer spesifikt linjene 34-38 (i min kode):

Et bilde som inneholder tekst, skjermbilde, Font, nummer

Automatisk generert beskrivelse

Dette setter lengden på meldingen i edx, meldingen selv i ecx, laster den standard output-filbeskrivelsen (1) inn i ebx for å spesifisere en standard utgang, og deretter setter SYS\_WRITE inn i eax som spesifiserer hvilket systemkall som skal tilkalles.

Når vi deretter kaller på 'int 80h', ser den at den har SYS\_WRITE (4) i eax, så sjekker den ebx for å se hvilken type utgang det skal være, ecx for å vite hvor meldingen er, og edx for å se hvor mange tegn ordet har, slik at systemet vet hvor slutten på meldingen er.

4)

Etter at linje 43 er utført, fortsetter programmet enten til linje 45 hvis inndataen er gyldig, eller hopper til linje 59/60 (som er Slutt:-etiketten) hvis inndataen er ugyldig.

5)

Strengt tatt i 'lessiffer:'-blokken står det bare 'push eax' og 'push ebx', men det er ingen kall eller hopp i 'lessiffer'-blokken, så hver gang du kaller på 'lessiffer' vil programmet også utføre 'Løkke:', som henter et tegn fra tastaturet, sjekker det og legger det i ecx.

Et bilde som inneholder tekst, skjermbilde, programvare

Automatisk generert beskrivelse

6)

Ret på linje 122 returnerer tilbake til stedet hvor 'lessiffer' ble kalt, enten 43 eller 50.

Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse

Koden returnerer så til stedet hvor 'lessiffer'-blokkene blir kalt, der programmet sammenligner edx og 0, som ikke vil være like siden 'Feil' legger 1 inn i edx for å signalisere en feilaktig lesning. Dette fører da til et hopp til 'Slutt'.