LAB5: Dataprogrammer

PUBLISERT: 2. mars 2015
FRIST: Tirsdag 10. februar 2015. Fristen er for å få en tilbakemelding og for å hjelpe dere til å strukturere arbeid. Den endelige innleveringen for mappeevaluering er i slutten av mai 2015.

HVIS NOE I DENNE OPPGAVEN ER UKLART, TA KONTAKT MED HJELPELÆRER eller LÆRER.

Kontakt hjelpelærer Christer Jonassen, chrisj13@student.uia.no.
Kontakt lærer Janis Gailis, janis.gailis@uia.no, kontor H1-036, intern tlf. (3814) 1562, skype konto: ocoss1530 (må avtales på foran ved å sende e-post).

Oppgave

Oppgaven bør hjelpe dere til å forstå måten et miljø for utførelse av dataprogrammer (les operativsystem) er designet på. Dere er allered litt kjent med kommandovinduet (shell, terminalvindu), som er et "vindu" inn i operativsystemet. Gjennom dette "vinduet" og ved hjelp av andre programmer (som python, for eksempel), kan man følge "liv" til et program fra den blir skrevet av dere, til den blir representert som en prosess i et datasystem.

1. OPPGAVE: lag et enkelt program i C og kompiler det med gcc
    a. OPPGAVE: i en teksbehandler skriv main() og lagre som test1.c
    b. OPPGAVE: kompiler programmet med gcc (se https://www3.ntu.edu.sg/home/ehchua/programming/cpp/gcc_make.html#zz-1.4)
    c. OPPGAVE: forklar hvordan man kan produsere test1.i, test1.s og test1.o og hva disse filene illustrerer
2. Åpne en Python shell for eksperimentering
    a. Åpne filen for å lese den bit for bit (eller byte for byte)
    b. Python har en module struct, som kan lese binære filer bit for bit; import struct (se https://docs.python.org/2/library/struct.html)
    c. f = open("test1", 'rb') (se https://docs.python.org/2/library/functions.html#open)
    d. OPPGAVE: finn ut hva betyr 'rb'
3. Prøv denne setningen bin = struct.unpack('B', f.read(1))[0]
    a. OPPGAVE: finn ut hva 'B' betyr
    b. OPPGAVE: finn ut hva f.read(1) gjør

   c. OPPGAVE: finn ut hva unpack funksjonen returnerer som resultat
   d. OPPGAVE: finn ut hvilken verdi bin har forutsatt at filen som du analysere er en binær “ELF 32-bit LSB executable”
4. Eksperimenter
   a. bin >> 7
   b. bin >> 7 & 1 (se https://wiki.python.org/moin/BitwiseOperators)
   c. str(bin >> 7 & 1) og det samme for 6,5, ⋯ 1,0 (se https://docs.python.org/2/library/functions.html#str)
   d. OPPGAVE: hva hvis bin >> 8?
   e. [str(bin >> x & 1) for x in (7,6,5,4,3,2,1,0)]
   f. ‘’.join([str(bin >> x & 1) for x in (7,6,5,4,3,2,1,0)])
   g. hex(bin)
   h. OPPGAVE: hva blir dette og hvordan er dette relatert til ELF? Begrunn.
   i. gjør punkt 3) andre gangen
   j. OPPGAVE: hva skjedde med verdien til bin? Begrunn
5. Sjekk ELF spesifikasjon og finn ut hvilken Python funksjon du kan bruke for å illustrere best, det som står i ELF spesifikasjonen
6. Gjør punkt 3) to ganger til og noter verdiene til bin
   a. OPPGAVE: sjekk denne artikkelen (se http://en.wikipedia.org/wiki/List_of_file_signatures) og forklar hvordan det henger sammen med det du har funnet
   b. OPPGAVE: hvilket navn/begrep brukes på det du har funnet etter å ha brukt f.read(1) 4 ganger?
7. Gjør kall til f.read(1) femte og sjette gang
   a. OPPGAVE: forklar verdiene
   b. OPPGAVE: Finn ut hvilken module og hvilken funksjon du kan bruke for å finne den samme verdien som du har i bin etter 6 kall til f.read(1).
8. OPPGAVE: Åpne et nytt terminalvindu (la python shell vinduet være åpent) og sjekk ut om din python prosess har åpne filer (se http://www.cyberciti.biz/faq/howto-linux-get-list-of-open-files/) Forklar.
9. OPPGAVE: hvordan ls -l /proc/<pid>/fd henger sammen med deler av innholdet i denne artikkelen (se http://en.wikipedia.org/wiki/File_descriptor) Forklar.
10. OPPGAVE: Prøv å finne ut hvordan du kan lukke fil i pyton shell, gjør det og sjekk igjen hvilke filer som python shell fortsatt holder fil-deskriptor til; Forklar.
11. Hvis filen er lukket, gjør punktene 2) til 6) på nytt, men bruk denne gangen en mindre bildefil i jpg, jpeg format. Forklar.
    a. Kan dere nå si noe om hvordan et miljø (OS) som gir mulighet til å utføre programmer, bør designes (i store trekk)?
    b. Hvordan hadde dere designet et python program som kunne utføre

en ELF fil?

Denne oppgaven er uavhenging og har som et formål å introdusere virkemåte for operativssytemet. Hvis dere ønsker å beherske programvareutvikling fult ut, er det viktig å forstå grunnleggende prinsipper bak kompilering, linking (statisk og dynamisk) og utførelse av programmet.

Ressurser

GCC and Make.Compiling, Linking and Building C/C++ Applications. (March, 2015). https://www3.ntu.edu.sg/home/ehchua/programming/cpp/gcc_make.html#zz-1.4

BUFFER OVERFLOW 4: A Compiler, Assembler, Linker & Loader. Retrieved March 2015 from http://www.tenouk.com/Bufferoverflowc/Bufferoverflow1c.html

Materiell (slides) lastet opp i Class Fronter og temaer gjennomgått på samlinger.

SLUTT.