SENSORVEILEDNING
PG3401 VÅR 2025

Linux

Et av de første kravene som testes med denne eksamen er evnen til å faktisk kjøre og bruke Linux. På årets eksamen kommer dette til uttrykk ved krav om å kjøre verktøyet EWA som er et verktøy studentene får utlevert og som kun støtter Debian 10, og er kompilert for Intel/AMD CPU (for studenter som bruker en Windows maskin med VmWare Workstation) og for ARM CPU (for studenter som bruker Mac med M1, M2, M3 eller M4 CPU).

En student som enten ikke kjører Linux I det hele tatt, eller som ikke er I stand til å starte et verktøy fra Terminal I Linux, vil ikke ha mulighet til å kjøre EWA verktøyet. Dette inkluderer studenter som har forsøkt å få ChatGPT eller annen generative AI til å svare på eksamen for seg. Er en student I denne kategorien vil studenten få F på eksamen, det spiller ingen rolle om studenten faktisk har levert kode som svarer på oppgavene. Emnet heter “C Programmering for Linux”, og har følgende læringsmål som måles direkte av dette krav:

- forstår hva "virtualisering" er
- kan operere kommandolinjen i linux for å utføre viktige operasjoner inkludert: ls, cd, mkdir, rm, cp, chmod, chown, pwd, ps, kill, man, wget og pipes, redirection og starte programmer og mounting av filsystemer
- kan forholde seg til linux som operativsystem

Gitt emnets navn er disse læringsmålene for emnet helt essensielle, og en student som ikke klarer å bruke Linux til enkle oppgaver kan ikke gis en bestått karakter.

Oppgave 1

1-2 poeng: En student har svart på en eller flere av oppgavene, men det er mye feil og det som er svart er veldig tynt.
3 poeng: Studenten har svart helt greit, men litt tynt, typisk under en side totalt.
4 poeng: Studenten har svart godt på oppgavene (men uten kildehenvisninger)
5 poeng: Et full-godt svar, med en eller flere kildehenvisninger.

Oppgave 2

Dette er ment å være en veldig enkel oppgave som tester basis kunnskaper hos eleven, frem til forelesning 5 med «input / output». De fleste studenter bør ha brukt forholdsvis kort tid på å løse denne, i tillegg er dette en oppgave som er ganske lik oppgave 2 på tidligere eksamener. 15 prosent er ganske mye uttelling, og full score på denne oppgaven og oppgave 1 er halvparten av kravet for å få bestått.

Hoveddelen av oppgaven er å lese en fil og opprette en output fil. Det trekkes cirka 50% hvis en student ikke har klart å opprette en fil korrekt. Videre gis det trekk i poeng for å ikke lukke filhandles riktig, eller minnelekkasje, for å ikke sjekke returkoder, for manglende kommentarer i koden, for å ikke opprette en header file for hver source fil, men for eksempel legger funksjonsprototypene øverst i C filen med main, for ikke lagre data i filen binært, men en eller annen kreativ tolkning, eller for veldig uryddig kodestil, feks ingen innrykk (tab/space).

Oppgave 3

Dette er en oppgave studentene bør være godt forberedt på å løse da den ligner veldig på en tilsvarende oppgave i oppgavesettet for eksamensforberedelse – og studentene får beskjed i forelesning om at dette er en viktig oppgavetype å være forberedt på. En student som har jobbet godt med øvingsoppgavene kan gjenbruke både UI og flyt i programmet, samt sin lenket liste implementasjon (cirka 80 prosent av oppgaven hvis studenten har skrevet godt strukturert og modulær kode).

Det gis poeng for å ha noe ekstra på denne oppgaven, for eksempel et UI grensesnitt som går utover det som er undervist i emnet (studentene får gjennom EWA verktøyet se hvordan det er mulig å lage et strukturert UI grensesnitt, og de beste studentene forventes å ha dratt inspirasjon fra dette), eller en modulær og fleksibel datamodell.

Videre gis det trekk i poeng for manglende cleanup kode ved exit, for dårlig kommentert kode som er tung å lese og har uoversiktlig struktur, for kode som har minnelekkasjer eller buffer overflow ved lesing av input, for globale variable (for eksempel head og tail pekere), for kode som IKKE bruker dobbelt-lenket liste, for hver funksjon som ikke er implementert etter oppgaveteksten, og hvis studenten har kode liggende i header filer eller at en C fil inkluderer andre C filer (altså istedefor å linke sammen kildefilene med makefile filen…)

Oppgave 4

Dette bør være en enkel trådoppgave som kun krever å forstå de grunnleggende elementene, erfaring i større og mer komplekse programmer med mange tråder som jobber i parallell vil være utenfor hva som kan forventes av studenter.

Oppgaven er todelt, og studenter som fullfører deler av første oppgave blir vist andre delen av oppgaven. Da ikke annet er beskrevet i eksamensteksten teller del 1 og del 2 like mange poeng. Det gis videre fratrekk i poeng for å mangle lokale variable i main funksjonen som sendes som parameter til trådene, for å ikke lese filnavnet fra kommandolinjen og sende til Tråd A, for å ikke ha endret til å bruke semaforer, for å ikke ha endret til eksplisitt initialisering (med *_init funksjonene), for manglende kommentarer i koden for å dokumentere hva koden gjør, og for å ikke sjekke returkoder og håndtere feilsituasjoner på en god måte.

Oppgave 5

Dette er en oppgave studentene bør være (overraskende) godt forberedt på. Studentene har fått nesten akkurat den samme oppgaven i oppgavesettet til eksamensforberedelse og studenter som har løst dette på en strukturert og dynamisk måte kan gjøre mindre modifikasjoner til koden for å få den til å passe med eksamensoppgaven. EWA verktøyet minner studenten på dette.

Det gis full score for en oppgave som løser hele oppgaven, det gis videre fratrekk for å ikke ha en god håndtering av brukerinput, for kode som ikke bruker structen i oppgaven som en skikkelig protokol (for eksempel leser og tester magic number og size før den aksesserer data), for kode som ikke mottar DYNAMISK svar (for filen som mottas, de andre kan være statiske strukter da størrelsen er fast), for eksempel går i en loop og leser mer data hvis det kommer mer, for hver seksjon av koden som ikke er ferdig (for eksempel kun har rukket å teste den første receive vil få trekk for hver påfølgende seksjon), og for kode ikke har klart å opprette korrekt output fil.

Oppgave 6

Dette er ment som en litt krevende oppgave for å kunne skille A og B studentene fra de andre karakterene. Det er et element av problemløsing i oppgaven i form av at studenten for det første selv må forstå at dataene som mottas er i en HTTP protokoll, og etterpå på studenten klare å dekryptere filen som er mottatt (ved hjelp av brute force, altså teste alle 256 mulige kombinasjonene av krypteringsnøkkelen). I tillegg er både portnummer og krypteringsnøkkel forskjellig for hver gang EWA starter oppgave 6, som gjør at studenten må skrive litt fleksibel kode og ikke forvente at det som mottas er likt som forrige kjøring.

Totalvurdering

Etter å ha poengberegnet alle oppgavene vil en total score bli satt. Endelig karakter vil kreve at sensor vurderer total-inntrykket av students arbeid og utvist forståelse for emne, samt poengmessig oppnåelse på oppgavene totalt.