

PG3401 EXAM 2025

Task 1. Theory (5 %)

A. Explain what the C programming language can be used for.

Programmeringsspråket C er et allsidig og effektivt programmeringsspråk som ofte blir omtalt som et "systemprogrammeringsspråk", ettersom det i stor grad benyttes til utvikling av kompilatorer, operativsystemer og annen systemprogramvare (Kernighan & Ritchie, 1988, s. 1). C brukes til alt fra enkel systemprogrammering til utvikling av systemer og programvare som krever høy ytelse og effektiv ressursbruk. På grunn av språkets effektivitet og fleksibilitet har C hatt en sentral rolle i utviklingen av programvareutvikling. Selv om C ikke tilbyr moderne abstraksjoner som objektorientering eller sømløs filhåndtering, gir dette også fordelene av enkelhet og stabilitet, noe som gjør C til et kompakt og lettlært språk med høy ytelse (Kernighan & Ritchie, 1988, s. 1 & 2).

Litteraturliste:

Kernighan, B. & Ritchie, D. (1998). The C Programming Language (2nd ed.)

B. Who is Dennis Ritchie and what is he known for in the field of Information Technology?

Dennis Ritchie var en amerikansk forsker innen informatikk. Han er mest kjent for å ha utviklet programmeringsspråket C og for å ha vært med å utvikle operativsystemet UNIX sammen med Ken Thompson (NSTMF, u.å.). Arbeidet til Dennis Ritchie la grunnlaget for store deler av programvareutvikling, dette inkluderer operativsystemer som Linux og macOS som er bygget på UNIX. «Unix, med sin modulære tilnærming, la vekt på enkelhet og effektivitet – prinsipper som fortsatt preger moderne systemer som Linux og macOS» (Quantum News, 2025). I tillegg skrev han boken The C Programming Language sammen med Brian Kernighan, som ble en viktig ressurs for utviklere over hele verden.

Litteraturliste:

National Science and Technology Medals Foundation. (u.å.). Dennis Ritchie.

Hentet fra: <https://nationalmedals.org/laureate/dennis-ritchie/>

Quantum News. (2025, 14. april). *Dennis Ritchie: The Forgotten Genius Who Built the Digital Age*. Quantum Zeitgeist. Hentet fra: <https://quantumzeitgeist.com/dennis-ritchie-the-forgotten-genius-who-built-the-digital-age/>

C. List at least 5 different Linux “distros”, explain briefly their background, if they have any specific purpose, and how they differ from each other.

1. Ubuntu

Ubuntu er en av de mest kjente Linux-distribusjonene, kjent for sin brukervennlighet og godt egnet for nybegynnere som ønsker å lære Linux (GeeksforGeeks, 2025). Det er en Debian-basert distribusjon utviklet av Canonical, og ble lansert i 2004 av Mark Shuttleworth sammen med et team tidligere Debian-utviklere. Målet var å skape en enkel og tilgjengelig Linux-distribusjon for vanlige brukere, kalt Ubuntu (Ubuntu, u.å.).

2. Debian

Debian er en av de eldste og mest stabile Linux-distribusjonene. «Debian er kjent for sin forpliktelse til stabilitet og prinsippene om fri programvare» (GeeksforGeeks, 2025). Debian har lagt grunnlaget for flere andre populære distribusjoner, blant annet Ubuntu og Linux Mint.

3. Fedora

Fedora er en moderne Linux-distribusjon sponset av Red Hat, kjent for sitt fokus på innovasjon og ny teknologi (GeeksforGeeks, 2025). Distribusjonen brukes ofte av utviklere og i akademiske miljøer fordi den støtter de nyeste versjonene av programvare og systemkomponenter. Fedora fungerer som en testplattform for Red Hat, noe som gjør den til en sentral komponent i Red Hats utviklingsmodell.

4. Arch Linux

Arch Linux er en “rolling release” distribusjon, noe som betyr at systemet oppdateres kontinuerlig i stedet for å ha faste oppdateringsversjoner (GeeksforGeeks, 2025). Den er rettet mot erfarne og teknisk dyktige brukere, og bygger på KISS-prinsippet (Keep It Simple, Stupid), som handler om å holde systemet enkelt uten unødvendig kompleksitet.

5. Linux Mint

Ifølge DistroWatch er Linux Mint en distribusjon basert på Ubuntu, den ble først lansert i 2006 av Clement Lefebvre, en fransk IT-spesialist bosatt i Irland (DistroWatch, u.å.). Linux Mint ble laget for som et mer brukervennlig alternativ til Ubuntu, med nødvendige programmer og kodeker forhåndsinstallert. I motsetning til systemer som for eksempel Arch Linux, krever ikke Linux Mint tekniske forkunnskaper, den er klar til bruk med en gang.

Litteraturliste:

Geeksforgeeks. (2025, 20. mars). *8 most popular Linux distributions*. Hentet 2. mai 2025 fra <https://www.geeksforgeeks.org/8-most-popular-linux-distributions/>

DistroWatch. (u.å.). *Major distributions*. Hentet 2. mai 2025 fra <https://distrowatch.com/dwres.php?resource=major>

Ubuntu. (u.å.). *The story of Ubuntu* Hentet fra: <https://ubuntu.com/about>

Task 2. File management and functions (15 %)

I denne oppgaven har jeg laget et program som leser en tekstfil og generer metadata basert på innholdet i filen. Programmet beregner en hash-verdi av filnavnet, finner størrelse og summerer tegnverdier, og teller forekomsten av hver bokstav. All info lagres i en struktur og skrives til en binær outputfil (pgexam25_output.bin). Alle krav er fullført, og programmet kompilerer og kjører uten feil.

Skjermbildene under viser at programmet kjøres som forventet:

```
Eastwill Assistant - EXAM 2025 v1 (id00-32007113)

Hi, task 2 is a good start on this exam, it is a simple I/O task,
to solve it you need to make a makefile to link several files together,
the application then needs to read a file, call the new functions on the
data and write a new file. If you want I can help to test the result.

I have created a testfile 'pg3401_exam25/task2_3/pgexam25_test.txt'
Use this file and the exam text to create your client application.

Ready to test the result? Press 1 to test, 0 to do it later: 0

Info: Exiting application...
Press [Enter] key to continue
Copyright Eastwill Security 2025
```

```
osboxes@osboxes: ~/Desktop/PG3401/EWA
File Edit View Search Terminal Help

Eastwill Assistant - EXAM 2025 v1 (id01-32009013)

You have asked me to check your result, let us have a look...

Successfully opened 'pg3401_exam25/task2_3/pgexam25_output.bin'.
File is 72 bytes large, so you did not use pragma pack - let me read it.
Successfully read the binary file - check
The hash in the struct is correct - check

I have not checked the code quality, that I will leave up to the
assessor, but I can read the file and it looks fine.
When you think your code quality is good enough this task is done.
Good work on this task - keep it up :)

Press [Enter] key to continue
Copyright Eastwill Security 2025
```



Task 3. List handling (20 %)

I denne oppgaven har jeg laget et system som håndterer flyavganger ved hjelp av en dobbeltlenket liste, der hvert fly objekt inneholder en enkel-lenket liste med passasjerer. Alle krav i oppgaveteksten er gjennomført. Programmet tilbyr totalt 10 funksjoner via et menybasert grensesnitt, inkludert mulighet for å legge til flights og passasjerer, endre seter, søke, slette og vise informasjon. Det er også mulig å få en full oversikt over alle flights og passasjerer. Applikasjonen kompilerer og kjører uten feil eller krasj. Skjermbildene nedenfor viser menyen ved oppstart og programmet i bruk.

```
osboxes@osboxes:~/Desktop/PG3401/EWA/pg3401_exam25/task3_3$ ./main
Flight Management Menu:
1. List all flights
2. Add new flight
3. Add passenger to flight
4. Change passenger seat
5. Delete a flight
6. Find flight by destination
7. Print flight details by index
8. Search passenger by name
9. Find passanger on multiple flights
0. Exit
Enter choice: █
```

```
osboxes@osboxes: ~/Desktop/PG3401/EWA/pg3401_exam25/task3_3
File Edit View Search Terminal Help
gcc -c -o obj/task3_menu.o task3_menu.c -O2 -I./include
gcc -o main obj/main.o obj/task3_menu.o obj/task3_flight.o obj/task3_passenger.o -O2 -I./include
osboxes@osboxes:~/Desktop/PG3401/EWA/pg3401_exam25/task3_3$ ./main
Flight Management Menu:
1. List all flights
2. Add new flight
3. Add passenger to flight
4. Change passenger seat
5. Delete a flight
6. Find flight by destination
7. Print flight details by index
8. Search passenger by name
9. Find passanger on multiple flights
0. Exit
Enter choice: 3
Flight index to add passenger to: 1
Name: Ola Nordmann
Seat: 72
Age: 29

Flight Management Menu:
1. List all flights
2. Add new flight
3. Add passenger to flight
4. Change passenger seat
5. Delete a flight
6. Find flight by destination
7. Print flight details by index
8. Search passenger by name
9. Find passanger on multiple flights
0. Exit
Enter choice: 1
Flight #1: SK103 -> OSLO | Seats: 120 | Time: 1030
Seat 72: Ola Nordmann (29 years)
Flight #2: DY456 -> BERGEN | Seats: 100 | Time: 1200
Flight #3: SK489 -> STAVANGER | Seats: 80 | Time: 1630
Flight #4: DA199 -> LONDON | Seats: 140 | Time: 1830
```

Task 4. Threads (20 %)

I denne oppgaven har jeg laget et tråd-basert program som leser en fil ved hjelp av to tråder, som deretter lager en hash-fil og en kryptert enc-fil. Det brukes en struktur kalt SharedData for å dele data mellom trådene, og synkronisering er brukt ved hjelp av semaforer og mutex, slik det er beskrevet i oppgaveteksten.

Resultatet lagres i en enc-fil. Alle krav i Part I og part II er fullført, og programmet kjører uten noen feil. Skjermbildene under viser at programmet fungerer som forventet:

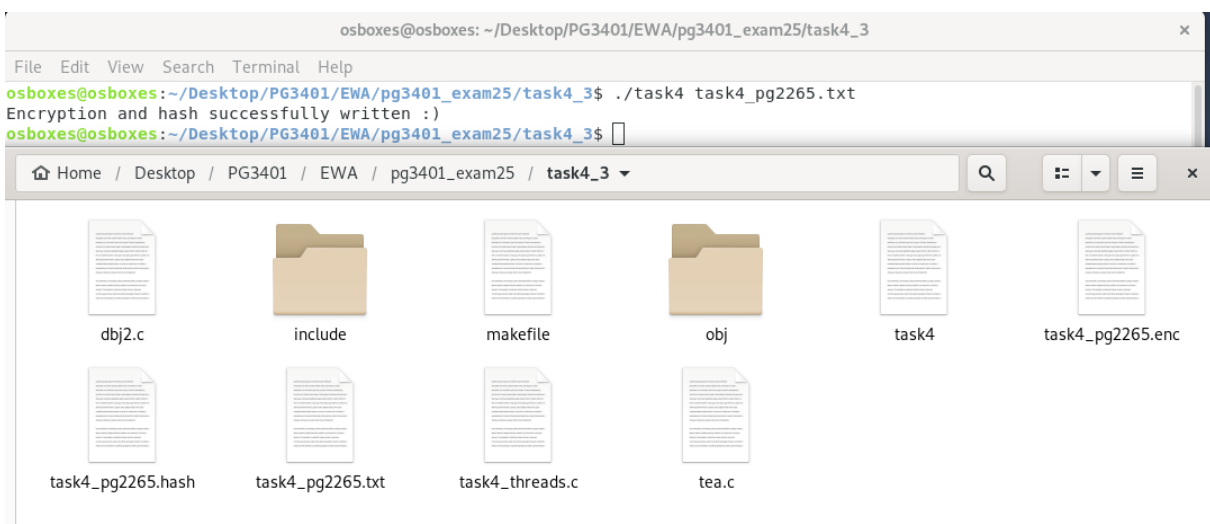
```
osboxes@osboxes: ~/Desktop/PG3401/EWA
File Edit View Search Terminal Help

Eastwill Assistant - EXAM 2025 v1 (id13-32005813)

You have asked me to check your result, let us have a look...
Successfully opened 'pg3401_exam25/task4_3/task4_threads.c' - check
Changes detected in file (hash e424740d)
Successfully read the source file - check
Successfully found a makefile in the task4 folder - check
Successfully found parameters to main - check
Successfully found semaphore variable - check
Successfully found explicit init - check
Successfully found 23 comments - check

Part I approved, press [Enter] to get Part II

Press [Enter] key to continue
Copyright Eastwill Security 2025
```



Task 5. Network (20 %)

I denne oppgaven har jeg laget en nettverksserver som bruker en enkel e-postprotokoll basert på ewpdef.h. Serveren starter ved å lese port og ID fra kommandolinjen, binder seg til 127.0.0.1 og venter på at en klient skal koble til. Når den er tilkoblet, håndteres meldingene i rekkefølge: HELO, MAIL FROM, RCPT TO, DATA og QUIT. Innholdet fra meldingen lagres i en .eml-fil sammen med avsender og mottaker.

Skjermbildene under viser at serveren starter som forventet og kommuniserer korrekt.

```
osboxes@osboxes: ~/Desktop/PG3401/EWA
File Edit View Search Terminal Help

Eastwill Assistant - EXAM 2025 v1 (id33-32008813)

Successfully connected to 127.0.0.1, port 2500
Successfully received: EWP0064|220 127.0.0.1 SMTP Smtptest 13 May 2025
Successfully sent: EWP0064|HELO bengt.127.0.0.1
Successfully received: EWP0064|250 127.0.0.1 Hello bengt.127.0.0.1
Successfully sent: EWP0064|MAIL FROM: <bengt@test.com>
Successfully received: EWP0064|250 Sender address ok
Successfully sent: EWP0064|RCPT TO: <student@kristiania.no>
Successfully received: EWP0064|250 Receiver address ok
#1 Successfully sent: EWP0064|DATA to_33-32008813.eml
#1 Successfully received: EWP0064|354 Ready for message
#1 Successfully sent: EWP0012|Test
#1 Successfully received: EWP0064|250 Message accepted
Successfully sent: EWP0064|QUIT
Successfully received: EWP0064|221 Server closed

Press [Enter] key to continue
Copyright Eastwill Security 2025
```

```
osboxes@osboxes: ~/Desktop/PG3401/EWA/pg3401_exam25/task5_3
File Edit View Search Terminal Help

osboxes@osboxes:~/Desktop/PG3401/EWA/pg3401_exam25/task5_3$ make
gcc -c -o obj/task5_server.o task5_server.c -O2 -I./include
gcc -o task5_server obj/task5_server.o -O2 -I./include
osboxes@osboxes:~/Desktop/PG3401/EWA/pg3401_exam25/task5_3$ ./task5_server -port 2500 -id Smtptest
Server is running on port 2500 with ID 'Smtptest'
Received HELO: bengt.127.0.0.1
Received MAIL FROM: <bengt@test.com>
Received RCPT TO: <student@kristiania.no>
Received DATA file: to_33-32008813.eml
osboxes@osboxes:~/Desktop/PG3401/EWA/pg3401_exam25/task5_3$ ls
include makefile obj task5_server task5_server.c to_33-32008813.eml
osboxes@osboxes:~/Desktop/PG3401/EWA/pg3401_exam25/task5_3$ cat to_33-32008813.eml
From: <bengt@test.com>
To: <student@kristiania.no>

Test
osboxes@osboxes:~/Desktop/PG3401/EWA/pg3401_exam25/task5_3$
```

Task 6. Problem solving (20 %)