



## KURSPLAN

# Programmering i UNIX-miljö

## Programming in UNIX Environment

7,5 högskolepoäng (7,5 ECTS credit points)

**Kurskod:** DV1457  
**Nivå:** Grundnivå  
**Fördjupning:** G2F  
**Utbildningsområde:** Teknik  
**Ämnesgrupp:** Datateknik

**Huvudområde:** Datavetenskap, Programvaruteknik  
**Version:** 11  
**Gäller från:** 2013-09-02  
**Fastställt:** 2013-05-21

### 1. Kursens benämning och omfattning

Kursen benämns Programmering i UNIX-miljö / Programming in UNIX Environment och omfattar 7,5 högskolepoäng. En högskolepoäng motsvarar en poäng i European Credit Transfer System (ECTS).

### 2. Beslut om fastställande av kursplan

Denna kurs är inrättad av Sektionen för datavetenskap och kommunikation 2013-05-21. Kursplanen har reviderats av Sektionen för datavetenskap och kommunikation och gäller från 2013-09-02.

Dnr: BTH 4.1.1-0274-2013

### 3. Syfte

Syftet med kursen är att studenten ska tillägna sig en fördjupad förståelse för UNIX-baserade system och för operativsystemnära programmering. Detta innebär bl a att kunna programmera på operativsystemets mest abstrakta nivå, närmast användaren, och nedåt genom abstraktionsnivåerna till den lägsta nivån, systemanropen. Kursen lär ut hur man designar mjukvara som interagerar med datorn via operativsystemet UNIX (och UNIX-baserade/liknande operativsystem såsom Linux och MacOS). Den lägger grunden för vidare studier inom andra områden (till exempel datasäkerhet) som kräver konkret förståelse för den underliggande teknologin. Syftet uppnås genom att studenten via en serie laborationer får praktisk erfarenhet av att utveckla program i en UNIX-miljö.

### 4. Innehåll

Kursen behandlar följande specifika abstraktionsnivåer av UNIX-programmeringen mer i detalj:

- Kommandon och script-språk. Scriptspråk är ett sätt att automatisera det som normalt anses vara användarinteraktion, det vill säga att interaktivt ge datorn kommandon att till exempel läsa in filer, sortera innehållet och skriva ut dem. På så vis är

script-språk (exemplifierat med Bourne shell) mycket kraftfulla, men tyvärr också mindre generella än vanliga programspråk.

- Programutveckling i C. C är grundpelaren i programmering av UNIX-applikationer, och ofta är stora delar av UNIX-system skrivna i C. Avancerad UNIX-programmering, t ex implementation av nätverkstjänster, utförs ofta i C med hjälp av system- och biblioteksanrop till operativsystemet.
- Programutveckling i assemblerspråk. Ibland är högnivåspråk såsom C inte tillräckligt uttrycksfulla eller bär på oönskad eller onödig funktionalitet, som kan påverka prestanda. Då kan det bli nödvändigt/viktigt att kunna tala "direkt" till datorn på dess egna språk, dvs i assemblerspråk. En viktig fördel med att kunna programmera "maskinnära" är en fördjupad förståelse för hur datorn verkligen arbetar.

### 5. Mål

#### *Kunskap och förståelse*

Efter genomförd kurs skall studenten:

- ha en förståelse för UNIX-systemets huvudsakliga delar och hur dessa delar hör ihop.
- ha grundläggande kunskap om programmering i skript-språk.
- ha en grundläggande kunskap om hur program skrivna i C interagerar med operativsystemet i en UNIX-miljö.
- ha grundläggande kunskap om programmering i assemblerspråk och programvarans interaktion på hårdvaran på en maskinnära nivå.

#### *Färdighet och förmåga*

Efter genomförd kurs skall studenten:

- kunna använda UNIX-systemets kommandoradgränssnitt.
- kunna skriva program med hjälp av byggstenarna i UNIX programmeringsgränssnitt.
- kunna utveckla program i C och assemblerspråk som kan köras i en UNIX-miljö.

*Värderingsförmåga och föhållningssätt*

Efter genomförd kurs skall studenten:

- kunna avgöra vilken abstraktionsnivå på programmeringen som är mest lämplig för att lösa en viss uppgift.
- muntligt och skriftligt kunna motivera, diskutera och värdera sina lösningar.

## 6. Generella förmågor

### 7. Lärande och undervisning

De teoretiska grunderna i kursen presenteras på föreläsningar och/eller övningar. Studenten förväntas dessutom att självständigt tillgodogöra sig teoretiska kunskaper med hjälp av självstudier av relevant litteratur.

De teoretiska kunskaperna tillämpas och fördjupas på praktiska laborations-/projektuppgifter där ämnesrelaterade problem ska lösas genom implementation i aktuellt system. Varje laboration redovisas skriftligt och muntligt. Engelska.

## 8. Bedömning och examination

### Examinationsmoment för kursen

Kod	Benämning	Omfattning	Betyg
1310	Shellscript-programmering	2.5 hp	A-F
1320	UNIX-programmering i C	2.5 hp	A-F
1330	Assemblerprogrammering	2.5 hp	A-F

Kursen bedöms med betygen A Utmärkt, B Mycket bra, C Bra, D Tillfredsställande, E Tillräckligt, FX Underkänd, något mer arbete krävs, F Underkänd. Slutbetyget är ett viktat och avrundat genomsnitt av betygen på momenten. Om sammanvägt betyg ligger exakt mellan två betygssteg sker avrundning nedåt.

## 9. Kursvärdering

Kursansvarig ansvarar för att studenternas synpunkter på kursen systematiskt och regelbundet inhämtas och att resultaten av utvärderingar i olika former påverkar kursens utformning och utveckling.

## 10. Förkunskapskrav

För tillträde till kursen krävs att den studerande har avklarat kurser i programmering, 15 hp, algoritmer och datastrukturer, 6 hp, operativsystem, 6hp och datakommunikation alt lokala nätverk, 4 hp.

## 11. Utbildningsområde och huvudområde

Kursen tillhör utbildningsområdet Teknik och ingår i huvudområdet Datavetenskap och huvudområdet Programvaruteknik.

## 12. Begränsningar i examen

Kursen kan inte ingå i examen med annan kurs, vars innehåll helt eller delvis överensstämmer med innehållet i denna kurs.

## 13. Övrigt

De huvudsakliga programmeringsspråken är C och Bourne Shell. Studenterna förväntas behärska dessa programmeringsspråk eller lära sig dem på egen

hand under kursens gång.  
Ersätter DV1305 och DV1425.

## 14. Kurslitteratur och övriga lärresurser

### Huvudlitteratur

- [1] Stephen Rago, Richard Stevens Advanced Programming in the UNIX Environment, 2nd ed. Addison-Wesley Professional, 2005, ISBN 0201433079
- [2] Material från sektionen.

--

### Referenslitteratur

- [1] Brian W. Kernighan, Dennis M. Ritchie The C Programming Language, 2nd ed. Prentice Hall, 1998, ISBN 0131103628
- [2] B. Stroustrup The C++ Programming Language, 4th ed. Pearson Education, 2013, ISBN 0-321-56384-0
- [3] N.M. Josuttis The C++ Standard Library: A Tutorial and Reference, 2nd ed. Pearson Education, 2012, ISBN 0-321-62321-52

