

1. Terminologi och funktion (4 p)

Sätt ett S (sant) framför de påståenden som är korrekta och ett F (falskt) framför de påståenden som är inkorrekta.

- [S] fstab används för att temporärt ansluta partitioner i filsystemet
- [S] Många Linuxdistributioner består till en stor del av program från GNU
- [S] Kommandot sed används för att byta ut ord mot andra ord i en text
- [S] Reguljära uttryck används av bash för filnamnsexpansion, t ex *.txt
- [F] Deadlock kan uppstå i ett system med en enda process (F)
- Kommandot grep används för att byta ut ord mot andra ord i en text (F)
- [S] I samband med sidväxling (paging) används algoritmen LRU (Least Recently Used)
- [F] Användarnas lösenord sparas nuförtiden i filen /usr/passwd
- [F] Programpaket i Ubuntu och Debian använder sig av apt-formatet
- [S] Den reguljära operatoren * matchar 0 eller flera av föregående tecken
- [S] Programpaket i CentOS använder sig av rpm-formatet (S)

- [F] I Unix-baserade operativsystem används alltid långa och förklarande namn på kommandon
- [F] En fil som har rättigheten 644 kommer alla användare i systemet kunna skriva till
- [S] Slackware och Gentoo är exempel på Linuxdistributioner
- [S] När ett program tilldelas minne, så vill vi undvika minnesfragmentering
- [F] Ett filnamn i ett Linux-system måste alltid ha en ändelse, som t.ex. minfil.txt
- [S] En användare i Linux kan öka prioritet på en av sina egna processer, för att den skall bli klar snabbare
- [F] Programmet "telnet" kan ge en säker anslutning till en server
- [F] Den reguljära operatoren ? matchar 0 eller flera av föregående tecken
- [S] Northbridge och Southbridge kan vara komponenter i en dator
- [S] Filnamnsexpansion i skalet, t ex *.txt, kallas ofta för "globbing"
- [F] Kommandot tail används för att byta ut ord mot andra ord i en text
- [F] Programpaket i CentOS använder sig av deb-formatet

Operativsystem:

Beskriv översiktligt ett operativsystems roll och huvudfunktioner. Varför behöver vi ett operativsystem?

Ett operativsystem är en programvaru- och systemmjukvarukomponent som fungerar som en mellanhand mellan användarens hårdvara och applikationsprogram. Dess huvudfunktioner inkluderar: Hantering av hårdvara, Hantering av filsystem, Processhantering, Säkerhet och Nätverkshantering.

Operativsystemet behövs eftersom det möjligt för flera användare att dela och använda datorresurser samtidigt och underlättar utveckling. Utan ett operativsystem skulle användare och program behöva hantera hårdvaran direkt, vilket skulle vara mycket mer komplicerat och ineffektivt.

Olika typer av datorsystem, som tex persondatorer, superdatorer, inbyggda system och smarta telefoner, kan behöva olika typer av operativsystem.

Nämn exempel på olika operativsystem och förklara vad som skiljer dessa operativsystem åt.

Persondatorer:

- Microsoft Windows: Ett vanligt operativsystem för persondatorer.

Smarta telefoner:

- iOS: Operativsystemet som används på Apple iPhones

Det som skiljer sig åt är användarupplevelse, tillgängliga applikationer och program, Säkerhet och integritet och Programvaruutveckling och stöd.

Ett operativsystem kan ha tre olika huvuduppgifter. Vilka är dessa uppgifter?

- Hantera maskinvara
- Tillhandahålla ett användargränssnitt
- Hantera och köra program

c) Redogör för de grundläggande skillnaderna mellan en monolitisk operativsystemskärna och en mikrokärna (struktur och fördelar/nackdelar). Rita en bild över hur strukturen ser ut mellan maskinvaran och användarprogrammen (user/kernel space). (4p)

- En monolitkärna är en kärna där alla operativsystemsfunktioner, såsom hantering av processer körs i kärnrymden som en stor och sammanhängande enhet.
 - **Fördel** : Snabbare kommunikation och enkelhet
 - **Nackdel**: Mindre skalbarhet och svårare underhåll
- En mikrokärna är en variant av lagerkärna där endast de mest grundläggande funktionerna hålls i kärnan.
 - **Fördel**: Ökar Säkerheten
 - **Nackdel**: Prestandaförlust

b) När ett operativsystem kör flera program samtidigt så körs de ofta inte äkta parallellt. Hur löser operativsystemet detta?

När ett operativsystem kör flera program samtidigt, särskilt på en enskild processor, sker inte äkta parallellitet eftersom en processor kan bara utföra en uppgift åt

gången. För att hantera flera program samtidigt använder operativsystemet tekniker som multitasking och virtuellt minne.

- Genom multitasking kan operativsystemet växla mellan olika program som körs på en dator.
- Genom att använda virtuellt minne kan operativsystemet låta flera program dela på fysiskt minne samtidigt.

Processer och trådar:

a) I ett flerprocesssystem så isoleras normalt processer från varann så mycket som möjligt. Om två processer vill kommunicera med varann, hur kan de då göra? (3p)

- processer kan kommunicera med varandra genom interprocesskommunikation (IPC).

b) I moderna system används ofta något som kallas "trådar". Beskriv kortfattat vad en tråd är, och vad den har för fördel jämfört med en process. (4p)

- En tråd är en mindre del av en process som kan utföra arbete parallellt med andra trådar inom samma process. Till skillnad från separata processer delar trådar samma adressrymd och andra resurser.

a) Förklara skillnaden mellan en process och ett program. Varför behövs processer?

Ett program är en samling instruktioner och data som är lagrade på lagringsmedia (till exempel hårddisk eller flashminne) och som är avsett att utföras av en dator.

Ett program är en samling instruktioner och data som är utformade för att utföra en specifik uppgift på en dator.

En process är en instans av ett program som körs på en dator. När ett program körs, laddas det in i minnet och en process skapas för att utföra programkoden.

Vi behöver processer eftersom processer är central del av operativsystem och behövs för flera skäl, till exempel processer möjliggör kommunikation och samarbete mellan olika program.

Processor:

- En processor är en kritisk komponent i en dator som utför instruktioner och hanterar data. Den fungerar som hjärnan i datorsystemet genom att utföra beräkningar.

Parent:

- Inom datavetenskap och operativsystem är "parent" (förälder) ett begrepp som används för att **beskriva förhållandet mellan två processer**, där en process skapar en annan process.

Minneshantering och Minne

Nästan alla moderna operativsystem använder sig av virtuellt minne.

a) Beskriv och förklara tre(3) problem som virtuellt minne löser. (3p)

Virtuellt minne löser flera problem i moderna operativsystem:

- Övergripande minneshantering : Virtuellt minne tillåter operativsystemet att hantera tillgång till fysiskt minne på ett effektivt sätt
- Minnes Fragmentering: Virtuellt minne minskar problemet med minnes fragmentering
- Minnes Överlappning och delning: Virtuellt minne möjliggör minnes överlappning och delning mellan olika processer och program.

Om ett system försöker få in mer processer i minnet än det finns RAM till, vad kan systemet göra då? Vad kan det få för effekter på systemet? (3 p)

- När ett system försöker få in fler processer i minnet än det finns fysiskt RAM tillgängligt kan det använda sig av virtuellt minne för att hantera detta scenario.

b) Vilka för- och nackdelar har virtuellt minnesteknik? (2p)

Fördel:

- Effektivare minnesanvändning
- Ökad flexibilitet:
- Bättre multitasking

Nackdel:

- Risk för page faults
- Prestandapåverkan

c) Ge exempel på vilka problem det skulle kunna ge om man kör ett operativsystem som tillåter många samtidiga processer och användare.

Om ett operativsystem tillåter många samtidiga processer och användare kan det leda till flera problem, inklusive Säkerhetshot, Förvirring och felaktig användning.

Beskriv olika lösningar till problemen i c) ovan. Visualisera gärna med bilder.

Säkerhetsåtgärder: För att skydda systemet och dess användare mot säkerhetshot kan operativsystemet implementera säkerhetsåtgärder som åtkomstkontroller, och antivirusprogram.

Kommandoskalet

a) `sudo bash`

- Skalet kör kommandot `sudo` för att få administrativa rättigheter
- Slutresultatet blir att du får ett nytt skal (bash) som körs med administrativa rättigheter.

b) `echo $PATH`

- Skalet ersätter `$PATH` med värdet av miljövariabeln `PATH`
- Slutresultatet blir att du ser sökvägarna som är inställda i miljövariabeln `PATH`

c) `rm -f /etc/shadow`

- Skalet kör kommandot `rm` för att ta bort filen `/etc/shadow`.
- Slutresultatet blir att filen `/etc/shadow` tas bort från systemet.

d) `cat hej.txt | grep [xyz] > tmp`

- Skalet läser innehållet i filen `hej.txt` med `cat`.
- Slutresultatet blir att rader från `hej.txt` som innehåller x, y eller z kopieras till filen `tmp`.

e) Vad är `stdin`, `stdout` och `stderr`?

- `Stdin` (standard input) är den ström av data som ett program läser in från.
- `Stdout` (standard output) är den ström av data som ett program skriver ut till.
- `Stderr` (standard error) är den ström av felmeddelanden som ett program genererar.

Vad gör kommandona `ls`, `pwd`, `whoami`, och `ps` ?

ls:

- Det visar en lista över filer och kataloger i den katalogen där kommandot körs.

pwd:

- Detta kommando skriver ut sökvägen till den katalog där användaren för närvarande befinner sig i terminalen.

whoami:

- Visar användarnamnet för den aktuella användaren som är inloggad på systemet.

ps:

- Visar en lista över aktiva processer som körs på systemet.

Filstruktur

6. Filstruktur (2p)

Säg att jag vill kopiera en fil från Andreas hemmakatalog till Thomas hemmakatalog. Sökvägen kan då anges antingen absolut eller relativt. Din nuvarande arbetskatalog är Andreas hemmakatalog. Hur ser då de två kopieringskommandoraderna ut?

_____ / _____ || Andreas Thomas

- Exempelfil

a) Kopiering med absoluta sökvägar (1p)

- `cp /hem/andreas/Exempelfil /hem/thomas/`

b) Kopiering med relativa sökvägar (1p)

- `cp Exempelfil ../../thomas/`

7. Filsystem (6p)

a) Beskriv följande ord utförligt: (2p)

- **Disk** : En disk är en lagringsenhet som används för att permanent spara data på en dator.
- **Partition** : En partition är en logisk uppdelning av en fysisk disk som används för att organisera och hantera data på lagringsenheten.
- **Filsystem** : Ett filsystem är en strukturerad metod för att organisera och lagra filer och kataloger på en lagringsenhet, såsom en disk eller partition.
- **Fil-träd** : Ett fil-träd, även känt som filsystemstruktur är en organisatorisk struktur som används för att representera filer, kataloger och deras relationer på en lagringsenhet.

För att systemet och en användare skall kunna komma åt sekundärminne eller lagring används montering och avmontering i operativsystem. Förklara processen för montering och avmontering i både Windows och Linux.

Glöm inte att ge utförliga svar!

a) Ge en beskrivning av vad montering och avmontering innebär i detta sammanhang. (1p)

- Montering och avmontering i detta sammanhang refererar till processen där ett filsystem, vanligtvis lagrat på en extern enhet som en hårddisk eller en USB-enhet, görs tillgängligt för systemet (montering) eller tas bort från systemet (avmontering).

b) Hur skiljer sig dessa processer åt i Windows jämfört med Linux-baserade OS? Ge exempel på kommandon eller verktyg som används i båda typerna av operativsystem. (3p)

- I Windows hanteras montering och avmontering av enheter automatiskt av systemet när en enhet ansluts eller kopplas bort från datorn. Exempel på kommandon och verktyg inkluderar `mountvol`,
- I Linux måste användare manuellt montera och avmontera enheter för att göra dem tillgängliga för systemet. För att montera en enhet i Linux används kommandot `mount`

c) Ett backsystem på 1000TB körs i ett system (med montering) som har en 100GB stor rotisk som är mer än halvfull. Varje dygn gör systemet en komplett backup på /home katalogen, som innehåller 10GB data. Hur påverkas systemet om monteringen tas bort?

Motivera ditt svar (2p)

- Om monteringen av backsystemets 1000TB-stora enhet tas bort skulle det inte påverka rotdisken (100GB stor) där systemet är installerat. Rotdisken är en separat enhet från backsystemet och är inte direkt beroende av det för att fungera.

Linux

Unix har en ganska rörig bakgrund, och många tidiga versioner skilde sig kraftigt från varann, så program och kunskaper var inte flyttbara mellan olika tillverkares Unix-varianter

a) Vad heter "standarden" som skapades för att försöka lösa det problemet? (1 p) När Linux hade funnits under några år började man se liknande problem med olika versioner där:

- Standarden som skapades för att försöka lösa problemet kallas "Single UNIX Specification" (SUS).

a) Vad heter "standarden" som skapades för att försöka lösa det problemet?

- "POSIX" (Portable Operating System Interface)

När Linux hade funnits under några år började man se liknande problem med olika versioner där:

b) Vad heter standardiseringen inom Linux? (1 p)

- "Linux Standard Base" (LSB).

De flesta Linux-baserade OS-distributioner kallar sig för GNU/Linux

c) Förklara kort vad GNU-projektet respektive Linux bidrar med till en typisk distribution?

- GNU-projektet bidrar med en stor del av programvaran och verktygen som behövs för att köra en typisk Linux-distribution.

d) Nämn en historisk person som har bidragit till Unix, Linux eller GNU och förklara vad denne har gjort. Max en(1) person får nämnas!

En historisk person som har gjort betydande bidrag till GNU-projektet är Richard Stallman. Han är känd för att vara grundaren av Free Software Foundation (FSF) och skaparen av GNU-projektet.

Scriptprogrammering och uttryck

a) Förklara vad följande bash-script gör. Beskriv funktionerna rad för rad. (4p)

```
#!/bin/bash // vilket skal som ska användas för att köra skriptet
if [ "$1" == "" ]; then // det är ett villkorssats som kontrollerar om det första argumentet är tom eller inte
    echo "Error!" // om villkoret i rad 2m är sant, kommer detta kommandot skriva "error"
    exit 1 // efter skriptet har skrivit ut felet, avslutas det med 'exit 1'
fi // avslutar if-satsen
mkdir $1 // om det första argumentet inte är tomt, skapas en ny katalog
cd $1 // kommandot används för att byta arbetskatalogen
for i in {1..100}; do // en for-loop som loopar 100, variabel "i" antar värden från 1 till 100.
    echo "Nummer: $i" >> nummer.txt // inom loopen skrivs textsträngen "Nummer:$i" till filen
    "nummer.txt". >> används för att lägga till texten i filen
done
```

Pipelining

a) Hur många processer skapas av kommandot? Motivera ditt svar! (1p)

`cat hitch.txt | uniq | sort -n | tail -n 3`

- Kommandot `cat hitch.txt | uniq | sort -n | tail -n 3` skapar en pipeline där flera kommandon körs. Varje kommando i pipelinen körs som en separat process. Således skapar detta kommando totalt fyra processer:

b) Vilka filnamn skriv ut av kommandot `ls *b*` om (1p)

kommandot `ls`` gav resultatet nedan:

abba hej.a hej.b boll.cpp boll.py Barbro.lön

- Filnamnen som skrivs ut av kommandot `ls *b*` är:
- hej.b
- boll.cpp
- boll.py
- Barbro.lön

-

c) Beskriv allmänt vad resultatet blir kommandot ``man find | grep ls``

- Resultatet av kommandot `man find | grep ls` blir en lista över alla avsnitt i manualsidan för kommandot `find` som innehåller ordet "ls".