

KTH ICT

Lab-PM

Operativsystem

Allmänna anvisningar

Laborations-PM

Operativsystem

1.0 Allmänna krav för godkänt

- Skriftlig rapport
- Fungerande program i enlighet med anvisningarna i uppgiftsbeskrivningarna
- För handledning och redovisning se kursinformation.

2.0 Minimikrav på skriftlig rapport

Rapporten ska vara maskinskriven. Vi rekommenderar någon ordbehandlare. Rapporten *skall* följa följande struktur, ytterliggare labspecifika krav på dokumentationen finns i vissa fall i labpecken

- **Framsida.** Tryckta framsidor bifogas. Fyll i alla uppgifter. Rapporter utan denna framsida kommer inte att rättas.
- **Problembeskrivning.** Beskriv kortfattat uppgiften.
- **Programbeskrivning.** Beskriv hur du valt att lösa uppgiften, speciellt val av datastrukturer och programstrukturen på den kod du har skrivit. Motivera också i berörda fall varför du har valt att lösa uppgiften på detta sätt. Inkludera en beskrivning av de funktioner du skrivit med respektive resultat- och argumentparametrar. Beskriv också om och hur du kan ändra beteendet, t. ex. byta ingående algoritmer, i ditt program.
- **Filkatalog.** Ange i vilken filkatalog (t.ex. `~pelie/os/lab1/`) du sparat din källkod och ditt program. Ange också vad dina filer heter.
OBS! Se till så att filerna är läsbara för assistenterna (se 3.4).
- **Utskrift med kompileringskommandon och köningar.** Bifoga en utskrift (t.ex. skapad med `script(1)`) som visar hur du kompilrat och kört ditt program. Speciellt viktigt är att du beskriver vilken kompilator du använt, alla använda kompileringsdirektiv och -flaggor eftersom vi som stickprov kommer att omkomplilera dina program. Visa några typiska testköningar. Visa såväl indata som utdata. När vi läser din rapport kommer vi att pröva dina exempel, samt naturligtvis våra egna.
- **Välkommenterad och strukturerad källkod.** *Okommenterad källkod accepteras inte!* Kommentarer skall beskriva väsentligheter i ditt program. Det är viktigt att du lägger ned litet extra möda på dina kommentarer för att snabba upp rättningsprocessen. I dokumentet CDOC i kurspärmens kan du få mer tips om konventioner för kommentarer i C. Om uppgiften endast inneburit förändringar/tillägg till existerande kod *skall* bara den kod du själv skrivit tas med i rapporten.
- **Verksamhetsberättelse och synpunkter på laborationen.** Beskriv arbetets utveckling. Hade du problem med verktygen, kompilatorn m.fl.? Hur lång tid har arbetet tagit? Skriv dina förslag till förändringar, idéer etc. Tyck fritt! Vi är angelägna om att få respons, så att vi kan förbättra till nästa år.
- **Genomförda tillägg.** Behövs endast om du gjort något förutom grundkraven.
- Uppskattning av tidsåtgång och eventuella kommentarer kring laborationen

3.0 Tips

Allt material som har att göra med kursen kan accessas via kurens hemsida:

<http://www.it.kth.se/courses/AKTUELLTKURSNUMMER> (t.ex ID2206)

3.1 Version av UNIX

Testexempel och program för laborationer är avsedda för exekvering under operativsystemet Linux på `shell.it.kth.se`. Man kan kontrollera vilken version av operativsystemet som körs m h a kommandot “`uname -a`”.

3.2 Process hantering

- PATH - då du startar en process letar kommandotolken (shell) efter angivet program i de kataloger som finns angiva i environment variabeln (envar) PATH. Undvik att ha med “.” i din PATH. Kontrollera din PATH med t.ex. `echo`:
 > `echo $PATH`
- Punkten ovan har en mycket viktig konsekvens, som förtjänar en egen punkt: När du har kompilerat ett program och skall köra det måste du skriva
 > `./mittprogram`
i stället för bara
 > `mittprogram`
eftersom kommandotolken inte tittar i “.”.
- Vill du veta var ett specifikt kommando ligger kan du ta reda på det m h a det inbyggda shell-kommandot `which`. Om man vill veta var kommandot `nm` ligger:
 > `which nm`

Observera att det kan även finnas flera versioner av ett och samma kommando. I flera UNIX/LINUX-installationer finns en uppsättning av GNU binärer utöver den standarduppsättningen som levereras med operativsystemet. Om GNU katalogen ligger först i din PATH kommer dess binärer att väljas av shell. I exemplet ovan kan det t.ex. hänta att `which` svarar:

/usr/ccs/bin/nm eller /opt/gnu/bin/nm
beroende på vilken katalog, `/usr/ccs/bin/` eller `/opt/gnu/bin`, som ligger först i PATH.

- Kontrollera vilka processer som du äger för tillfället:
 > `ps -a`

Genom att ge andra flaggor (switchar) till `ps` kan du få mer information om dina eller andras processer.

- Döda “runaway”-processer eller processer som hamnar utanför din kontroll. Det enklaste sättet att döda dem är m.h.a kommandot
 > `kill -KILL process-pid`
där `process-pid` är processens identifikationsnummer som kan fås t.ex. via kommandot “`ps -a`”.

3.3 Manualsidor

I UNIX/LINUX finns en stor del av dokumentationen maskinläsbar i så kallade manualsidor. Kommandot för att hämta en manualsida heter `man`. Manualsidorna är uppdelade i flera sektioner, och varje sektion täcker olika områden. För att hämta dokumentationen om systemanropet `close()` från sektion 2 skriver `man`:

> `man 2 close`

Det är inte nödvändigt att ange sektionsnumret innan funktionen, men det är säkrast, eftersom

man söker igenom sektionerna i nummerordning. Följande exempel visar vad som kan hända:

```
> man write
      write(1)          User Commands          write(1)
      NAME
      write - write to another user...
```

Det är inte den rätta manualsidan för systemanropet `write`. (1) betyder i sektion 1. Man skriver ofta sektionsnumret inom parentes efter funktionsnamnet. Manualsektionerna innehåller följande (detta kan ses om man använder `xman`):

```
1 - USER COMMANDS
2 - SYSTEM CALLS
3 - C LIBRARY FUNCTIONS
4 - FILE FORMATS
5 - HEADERS, TABLES AND MACROS
6 - GAMES AND DEMOS
7 - SPECIAL FILES
9 - DDI AND DKI
1 - LOCAL
```

Letar du efter ett speciellt kommando eller någon rutin kan du göra på fyra sätt:

1. Använd sökfunktionen i `xman`: öppna en manualsida och välj *search* under *optionsmenyn*.
2. Använd kommandot `apropos`:

```
> apropos lseek
llseek           llseek (2)      - move extended read/write file pointer
lseek            lseek (2)      - move read/write file pointer
```

Tyvärr brukar den lista som produceras innehålla en massa ovidkommande anrop. Ett bra tips är att "pipe:a" output till ett `grep`-kommando som filtrerar bort onödig information.

3. Du kan även läsa `Intro`-sidan i den aktuella sektionen:

```
> man 3 Intro
```

Denna sida innehåller en hel del ytterligare information om relaterade sektioner.

4. Hittar du inte vad du letar efter, kolla din **environment variabel** `MANPATH`. Den anger den lista av kataloger som kommando `man` letar i under sökningen.

3.4 Att byta filskydd.

- Använd kommadot `chmod` för att kunna reglera tillgängligheten till en fil eller katalog. Villkorflaggan `-R` tillåter rekursiv ändring.

- Ligger katalogen under AFS filsystemet måste du använda "`fs`" kommandot.

```
> fs listacl katalognamn
- visar åtkomstlistan för den givna filkatalogen. listacl kan förkortas la.
> fs setacl katalognamn användare/grupp rättigheter
- ger en viss rättighet åt en viss användare/grupp. setacl kan förkortas sa.
> fs setacl katalognamn användare/grupp none
- återkallar alla rättigheter för en viss användare/grupp.
> fs setacl katalognamn rron rl
- ger läs och list rättigheter för gruppen rron på era labbkataloger.
> fs help eller pts help
- visar alla kommandon i fs- eller pts-sviten.
```

Har du problem med att byta filskydd kan det bero på att du inte kör rätt `fs` kommando. Det finns flera `fs` kommandon och vilket du kör beror på vilket kommando som först hittas via din sökväg. Prova i så fall att köra kommandot `module add afs` innan du ändrar rättigheterna

KTH ICT

Framsida för labrapport

Operativsystem, ID220

Period _____, läsår 20_____

Fyll i alla uppgifter!

Labnr.	Labnamn	
<input type="text"/>	<input type="text"/>	
Efternamn, förmann	Personnummer	<i>Tydlig</i> datorpostadress
<input type="text"/> <input type="text"/>		
Inlämningsdatum <input type="text"/>		

Kommentarer

För internt bruk

Godkänd

Komplettera

Meddelad

Datum

Signatur

Registrerad

Ny

Gammal

G/B/U