Tentamen

Systemnära Programmering, 5p
25 September 1992

- Skrivtid 9–15
- **En** av följande böcker:
  - Kay & Kummerfeld: "*C Programming in a UNIX^{TM} Environment*"
  - Waite, Prata: "*Programmering i C*"
  - Kernighan, Ritchie: "*The C Programming Language*"
  - Korpela, Larmela: "*C-ohjelmointikieli*"
  - Atkinson & Atkinson: "*Using Borland C++*"
- Maxpoäng 48 (för godkänt krävs 24p, väl godkänt 36)
- Uppgifterna är slumpmässigt ordnade.
- Börja varje uppgift på *nytt* blad och skriv uppgiftsnummer längst ut till höger på bladet.
- Skriv namn på *varje* blad.
- Fråga om någonting är oklart!
- Återlämning (+ ev genomgång) av tentan sker onsdagen den 30/9 kl. 13.15 i S302.
- Resultatlista kommer upp på anslagstavlan så snart som jag har rättat klart.

Lycka Till!

Uppgift 1 (5 + 5 p)

(a) Vad skrivs ut av följande program? Markera också vad de enskilda utskrifterna skriver ut.

```
char *c[] = {"ENTER", "NEW", "POINT", "FIRST"};
char **cp[] = {c+3, c+2, c+1, c};
char ***cpp = cp;

main()
{
printf("\n%s", **++cpp);
printf("%s ", *--*++cpp+3);
printf("%s", *cpp[-2]+3);
printf("%s\n", cpp[-1][-1]+1);
}
```

(b) Rita en bild av datastrukturerna a, p och pp som de ser ut innan programmet körs. Vad skrivs ut av programmet?

```
#define PRINT(v1,v2,v3) \
printf("pp-p = %d\t*pp-a = %d\t**pp = %d\n", v1, v2, v3)

#define ett pp-p
#define tva *pp-a
#define tre **pp
int a[] = {0, 8, 12, -3, 4};
int *p[] = {a, a+1, a+2, a+3, a+4};
int **pp = p;

main()
{
pp++; PRINT(ett, tva, tre);
*pp++; PRINT(ett, tva, tre);
*++pp; PRINT(ett, tva, tre);
++*pp; PRINT(ett, tva, tre);
}
```

Uppgift 2 (4 + 3 + 2 p)

(a) Skriv en funktion `least(f1, f2, x)` i C som tar tre parametrar, två pekare till funktioner som returnerar double och en double. Funktionen ska returnera en pekare till den funktion som applicerad på x ger det minsta värdet.

Exempel: om anropet är `least(sin, cos, 0.0)` så ska resultatet vara en pekare till *sin* eftersom $sin(0.0) = 0.0 < cos(0.0) = 1.0$)

(b) Skriv ett macro `LEAST(f1, f2, x)`, som utför samma sak som i uppgiften ovan.

(c) Vad är det för skillnad på lösningarna? (dvs function vs macro) Ger de alltid samma resultat? Motivera dina svar.

Uppgift 3 (8 p)

Skriv ett program `circulate` (i C) som roterar bokstäver x antal steg. Om man roterar 3 steg ska alltså a bli d, b blir e, osv ända till y som blir b och z som blir c, dvs man tolkar alfabetet som en cirkel. Stora bokstäver ska förbli stora och små fortsätta vara små.

Programmet ska läsa från standard input (till EOF) och skriva på standard output.

`circulate [-n]`

där n är antalet steg som bokstäverna ska roteras. $0 \leq n \leq 26$, om n inte anges ska 1 steg användas. Observera att det endast är bokstäver (a-z och A-Z) som ska roteras, de övriga tecknen ska vara oförändrade. Följande rad

Hej, zombie!

ska alltså ge (med anrop `circulate` eller `circulate -1`)

Ifk, apncjf!

Programmet ska kontrollera att anropet är korrekt och att n ligger i rätt intervall.

Uppgift 4 (3 + 3 + 1 p)

(a) Vilka är de tre normala tillstånden en process kan befinna sig i? Tala kort om (tex med ett ord) vad processerna sysslar med i de olika tillstånden.

(b) Hur kan processer växla mellan de olika tillstånden? Åskådliggör med bild. Vad kan övergångarna bero på?

(c) Var kommer nya processer in? (Till vilket tillstånd.)

Uppgift 5 (4 p)

Vad menas med termen *short-term scheduler*? Beskriv en algoritm för en *short-term scheduler*, hur den fungerar, dess för- och nackdelar.

Uppgift 6 (4 + 6 p)

(a) Beskriv hur semaforer fungerar. Vad används de till? Vilka operationer på semaforer finns det, och hur fungerar de?

(b) Beskriv, med hjälp av text och pseudokod, hur man skulle kunna implementera generella semaforer med en monitor. (Om det skulle behövas så får du förutsätta att de vanliga operationerna på köer, stackar och länkade listor redan finns implementerade.)