Operativsystem (ID2200/06)
2017-01-XX XX:00-XX:00

Namn: _____

Instruktioner

- Du får, förutom skrivmateriel, endast ha med dig en egenhändigt handskriven A4-sida med anteckningar. Mobiler etc skall lämnas till tentamensvakterna.
- Svaren skall lämnas på dessa sidor, använd det utrymme som finns under varje uppgift för att skriva ner ditt svar.
- Svar skall skrivas på svenska eller engelska.
- Du skall lämna in hela denna tentamen.
- Inga ytterligare sidor skall lämnas in.

Versioner

Om detta är din ordinarie tentamen i ID2200/06 så skall frågorna 1-9 besvaras men inte frågorna under 10.

Om detta är en omtentamen för den version av kursen som gick före HT2016 så gäller följande:

- För de som läste ID2200 VT2016 är tentamen på 6hp och frågorna 1-9 skall besvaras.
- För de som läste ID2200 före VT2016 är tentamen på 3.8hp och endast frågorna 1 till 8 skall besvaras.
- För omtentander på ID2200 så gäller även att om det är så att du inte har fått godkänt på de laborationsmoment som kursen innehöll kan du välja att besvara frågorna under 9 och då få det momentet tillgodoräknat.
- För de som läste kursen ID2206 före HT2016 är tentamen på 4.5hp och endast frågorna 1 till 8 och 10 skall besvaras.
- För omtentander på ID2206 så gäller även att om det är så att du inte har fått godkänt på de laborationsmoment som kursen innehöll kan du välja att besvara frågorna under 9 och då få det momentet delvis tillgodoräknat. Frågorna under 9 motsvar ca: 2.2hp.

Betyg för 6hp

Tentamen har ett antal uppgifter där några är lite svårare än andra. De svårare uppgifterna är markerade med en stjärna, *poäng**, och ger poäng för de högre betygen. Vi delar alltså upp tentamen in grundpoäng och högre poäng. Se först och främst till att klara de grundpoängen innan du ger dig i kast med de högre poängen.

Notera att det av de 42 grundpoängen räknas bara som högst 38 och, att högre poäng inte kompenserar för avsaknad av grundpoäng. Gränserna för betyg är som följer:

- Fx: 22 grundpoäng
- E: 24 grundpoäng
- D: 30 grundpoäng
- C: 34 grundpoäng
- B: 38 grundpoäng och 12 högre poäng
- A: 38 grundpoäng och 18 högre poäng

Gränserna kan komma att justeras nedåt men inte uppåt. Gränserna är naturligtvis annorlunda för de som skriver en av de mindre versionerna av denna tentamen.

Gränserna är naturligtvis annorlunda för de som skriver en av de mindre versionerna av denna tentamen.

Erhållna poäng

Skriv inte här, detta är för rättningen.

Uppgift	1	2	3	4	5	6	7	8	9	Σ
Max G/H	4/0	4/2	4/2	4/2	4/2	4/2	4/2	2/2	12/8	42/22
G/H										

Totalt antal poäng:

Betyg:

Namn:_____

1 Operativsystem

1.1 Kommandon [2 poäng]

Ge en kort beskriving av vad kommandona nedan gör.

- `ls`
- `cd`
- `less`
- `rm`

1.2 Utdata [2 poäng]

Om vi ger kommandosevensen nedan, vad kommer vi då få för reslultat?

```
> echo "cd foo grep bar" | wc -w
```

2 Processer

2.1 Begränsad direkt exekvering [2 poäng]

Kortfattat beskriv vad som menas med "begränsad direkt exekvering" (linited direct execution).

Namn:_____

2.2 Vilket segment [2 poäng]

Vilka variabler i koden nedan är allokerade på stacken (även om de har i verkligheten kanske endast ligger i register).

```
#define MAX 4

int x = 43;

int foo(int a) {

  int *r = malloc(MAX * sizeof(int));

  for(int i = 0; i < MAX; i++) {
     r[i] = x + (a*i);
  }

  int z = r[0] + r[MAX];

  return z;
}
```

2.3 Ett system anrop [2 poäng*]

Beskriv kortfattat vad som händer när en användarprocess gör ett systemanrop.

Namn:_____

3 Schemaläggnig

3.1 First come first serve [2 poäng]

Antag att vi har tre job som tar 10ms, 10ms och 30ms var. Om schemalägger dessa jobben direkt efter varandra vad är då den bästa respektive sämsta medelvärdet för jobbens omloppstid (turnaround time)?

3.2 Shortest time-to-completion first [2 poäng]

Schemaläggaren "shortest time-to-completion first" är optimal men förutsätter att vi vet hur lång tid varje jobb kommer att ta. Detta är sällan något vi vet i förhand så varför är det fortfarande intressant se vad en schemalägging skulle ge med den stategin?

3.3 Svarstid [2 poäng*]

Om vi antar att vi har en schemaläggare som använder sig av "Round-Robin" och en tidsperiod på 10ms vid schemaläggning av processer. Hur skulle svarstiden förändras om vi halverar tidsperioden till 5ms, vad skulle vi få som en bieffekt?

Namn:_____

4 Virtuellt minne

4.1 Segmentering [2 poäng]

Vid implementering av segmentering behövs två värden för att beskriva ett segment, vilka är dessa värden?

4.2 Paging [2 poäng]

Antag att vi har en virtuell adress som består av ett 20-bitars sidnummer och en 12-bitars offset. Vi har en fysisk adressrymd på 32 bitar, kodar ett ramnummer i 20 bitar och element i en sidomvandlingstabell är fyra bytes stort. Hur stor är en sida och hur stor skulle en komplett omvandlingstabell behöva vara?

4.3 Inverterade sidtabeller [2 poäng*]

Hur många element måste vi ha en inverterad sidtabell och vad måste varje element innehålla?

Namn:_____

5 Minneshantering

5.1 malloc och free [2 poäng]

Procedurerna `malloc()` och `free()` är implementerade i bibliotek istället för som systemanrop, beskriv en fördel som detta medför.

5.2 Best fit [2 poäng]

En strategi vid allokering av minne är att ta det block som passar bäst s.k. "best fit". Vad är fördelen med denna metod och vad är nackdelen?

5.3 Trådspecifik malloc [2 poäng*]

Vad skulle fördelen vara att låta varje tråd i en process ha sina egna fria block som `malloc()` hade att välja bland?

Namn:_____

6 Flertrådad programmering

6.1 Trådar i en process [2 poäng]

Trådar i en process delar många saker men det finns en mycket viktigt datastruktur som de inte delar - varje tråd har sin egen; vilken är denna datastruktur?

6.2 Deadlock [2 poäng]

Beskriv en strategi för att hantera lås som gör att vi kan undvika deadlock i en multi-trådad beräkning.

6.3 Petersens algoritm [2 poäng*]

Beskriv vad Petersens algorithm skall lösa och varför den inte går att använda på de flesta moderna datorsystem.

7 Filsystem

7.1 Länkar [2 poäng]

Vad är skillnaden mellan en *hård* och en *mjuk* länk?

Namn:_____

7.2 inode [2 poäng]

Kortfattat - vad inehåller en *inode*?

7.3 Varför inte [2 poäng*]

I Unix får kan vi inte skapa flera hårda länkar till en *map* (*directory*). Varför har vi den begränsningen?

8 Virtualisering

8.1 Effektivitet [2 poäng]

När vi kör ett helt operativsystem virtualiserat så kommer exekveringen ta längre tid eftersom en vi då har en virtualisering i två nivåer. Ungefär hur mycket långasammare kommer ett beräkningsintensivt program att gå: nästan lika snabbt, halva hastigheter eller typiskt en faktor tio långsammare?

Namn:_____

8.2 Containers [2 poäng*]

Såp kallade "Linux containers" är också ett sätt att köra flera operativsystem på samma maskin. Vilken begränsning har denna metoden jämfört med full virtualisering?

9 Implementering

9.1 A knife a fork .. [2 poäng]

Givet programmet nedan, vad kommer att skrivas ut på skärmen?

```
int main() {
  int pid;
  int x = 0;

  pid = fork();

  if(pid == 0) {
    printf("  child:  x is  %d \n", x);
    x = 42;
    sleep(2);
    printf("  child:  x is %d  \n", x);
  } else {
    sleep(1);
    printf("  mother: x is  %d \n", x);
    x = 13;
    sleep(2);
    printf("  mother: x is %d  \n", x);
    wait(NULL);
  }
  return 0;
}
```

Namn:_____

9.1.1 Zombie [2 poäng*]

Förklara kortfattat vad en zombie är och vad det har med koden nedan att göra.

```
wait(&res);
printf("the result was %d\n", WEXITSTATUS(res));
```

9.2 Signaler [2 poäng]

9.2.1 Sigaction [2 poäng]

I koden nedan finns en referens till `gurka`, vad torde gurka vara och vad är egentligen syftet med koden?

```
struct sigaction sa;

sa.sa_handler = gurka;

sigemptyset(&sa.sa_mask);

assert(sigaction(SIGINT, &sa, NULL) != 0);
```

9.2.2 IDT [2 poäng]

Vilken roll spelare den s.k. *Interrupt Descriptor Table* vid signalhantering?

Namn:_____

9.2.3 Mellan processer [2 poäng*]

Signaler kan operativsystemet använda för att meddela sig med en användarprocess. Kan en användarprocess skicka en signal till en annan användarprocess? Finns det begränsningar?

9.3 Minneshantering

9.3.1 Ett steg tillbaka [2 poäng]

Koden nedan är från proceduren `free()` i en minneshanterare. För att frigöra det minne som pekas ut av `memory` så görs en väldigt konstig operation `memory -1`, vad är det man vill åstadkomma här?

```
void free(void *memory) {
  if(memory != NULL) {
    struct chunk *cnk = (struct chunk*)((struct chunk*)memory - 1);
    cnk->next = flist;
    flist = cnk;
  }
  return;
}
```

9.3.2 strace [2 poäng]

Du har just implementerat din första minnesallokerarer och den ser inte ut att konsumer för mycket minne. Du gör ett enkelt test nedan för att få lite bättre förståelse för hur hanteraren beter sig. Vad är det du vill veta?

```
> strace ./bench 2>&1 > /dev/null | grep brk | wc -l
```

Namn:_____

The number of calls to the system call `brk()`, less is better

9.3.3 brk() [2 poäng*]

Hur kan systemanropet `brk()` användas för att frigöra minne som operativsystemt allokerat åt en process?

9.4 Kommunikation

9.4.1 En namngiven pipe [2 poäng]

Du kan skapa en namngiven *pipe* genom att använda library proceduren `mkfifo()`. Vilket system används för att registrera och hitta namngivna `pipes`?

9.4.2 Pipe eller socket [2 poäng]

Om man har satt upp en *pipe* mellan två processer så kan man naturligtvis skicka meddelanden från A till B. Om man istället sätter upp en *socket* så kan man göra lite mer, vadå?

Namn:_____

9.4.3 Socket adressfamiljer [2 poäng*]

Vad är det för skillnad mellan `AF_UNIX` och `AF_INIT`?

9.5 Lagring

9.5.1 O_DSYNC [2 poäng]

Om vi kör tester för att räkna på hur snabbt vi kan skriva till en fil så får vi väldigt olika resultat beroende på om vi ger flaggan `O_DSYNC` eller inte. Vad har denna flagga för betydelse, hur skilljer sig resultaten och varför?

```
int flag = O_RDWR | O_CREAT | O_DSYNC;
int fd = open(name, flag, mode);
```

9.5.2 prestand [2 poäng]

Det tar några ns att läsa eller skriva till minnet, hur lång tid tar det att läsa från en fil som ligger på en hårddisk? Är det skillnad på om det är första gången man läser jämfört med om man läser filen för andra gången?

Namn:_____

9.5.3 tmpfs [2 poäng*]

Om vi monterar ett `tmpfs` filsystem så får vi bättre prestanda. Filsystemet har dock en nackdel som vi kanske inte alltid kan accepterar, vilken?

10 Bara för omtentamen i ID2206

Denna del är endast för de som gör omtentamen i ID2206 från tidigare år dvs före HT2016. Frågorna kommer att röra de kursmål som fanns med i den tidigare kursbeskrivningen men som har en mindre central roll i den nya kursbeskrivningen: realtids-operativsystem, distribuerade operativsystem och distribuerade filsystem.

10.1 Distribuerat filsystem

10.1.1 Vad kan detta vara [2 poäng]

10.1.2 Men detta då [1 poäng*]

10.2 Realtidsegenskaper

10.2.1 Rätt så lätt [2 poäng]

10.2.2 Klurigare [2 poäng*]