Systemutveckling i Python

Slutuppgift – Övervakningsapplikation

Syftet med slutuppgiften är att utveckla en övervakningsapplikation i Python som samlar in information från operativsystemet och presenterar denna information för användaren via en konsolbaserad meny.

Användaren ska kunna interagera med applikationen genom menyn för att få information om:
- CPU-användning
- Minnesanvändning
- Diskanvändning

Observera: När användaren interagerar med applikationen via konsolmenyn ska inga konfigurerade larm aktiveras.

Krav för Godkänd Nivå

När applikationen startas ska användaren presenteras med en meny med fem alternativ:

1. Starta övervakning

Startar övervakning av:
- CPU-användning
- Minnesanvändning
- Diskanvändning

Ingen övervakning ska starta automatiskt vid programstart. Användaren måste aktivt välja att starta övervakningen.

2. Lista aktiv övervakning

Visar information om aktiv övervakning och nuvarande status.

Om ingen övervakning är aktiv:
"Ingen övervakning är aktiv."

Om övervakningen är aktiv ska aktuell status visas, exempelvis:
CPU Användning: 35%
Minnesanvändning: 65% (4.2 GB av 8 GB används)
Diskanvändning: 80% (400 GB av 500 GB används)

Efter detta:
"Tryck Enter för att bekräfta och återgå till huvudmenyn."

3. Skapa larm

När användaren väljer detta alternativ visas en undermeny för att konfigurera larm:

Konfigurera larm:
1. CPU-användning
2. Minnesanvändning
3. Diskanvändning
4. Tillbaka till huvudmeny

Efter att användaren valt ett alternativ ska denne kunna ange en procentuell nivå där larmet ska aktiveras:
"Ställ in nivå för larm (mellan 0–100):"

Exempel:
Om användaren väljer 80, visas:
"Larm för CPU-användning satt till 80%."

Om användaren matar in något ogiltigt (t.ex. text eller tal utanför intervallet 1–100), ska ett felmeddelande visas:
"Fel: Ange ett tal mellan 1 och 100."

Efter bekräftelse återgår användaren till huvudmenyn.

Flera larm av samma typ ska kunna konfigureras.

4. Visa larm

Listar alla konfigurerade larm, sorterade efter typ.
Exempel:
1. CPU-larm 70%
2. Disklarm 95%
3. Minneslarm 80%
4. Minneslarm 90%

Efter detta:
"Tryck Enter för att bekräfta och återgå till huvudmenyn."

5. Starta övervakningsläge

Startar övervakningsläget, där applikationen kontinuerligt kontrollerar systemets status och jämför mot konfigurerade larmnivåer.

Vid start visas:
"Övervakningsläge är startat. Övervakning är aktiv, tryck på valfri tangent för att återgå till menyn."

Under pågående övervakning skrivs status ut med jämna mellanrum. Om ett larm aktiveras visas:
"*** VARNING: LARM AKTIVERAT – CPU-användning överstiger 80%! ***"

När användaren trycker på valfri tangent avbryts övervakningsläget och huvudmenyn visas igen.

Icke Funktionella Krav på Applikationen för Godkänd Nivå

• Programmet ska bestå av minst ett antal filer med kod som aktivt används. Dvs. all kod ska inte vara skriven i en fil.

• Programmet ska använda sig av objekt där det passar.

• Programmet ska vara skrivet med funktioner.

• Programmet ska innehålla funktionell programmering på minst ett ställe, t.ex. vid sortering av larm innan visning.

• Koden ska vara välskriven, dvs. lättförståeliga variabelnamn och funktionsnamn, kommentarer där det passar och en bra struktur.

• Koden ska vara bugfri. Funktionaliteten som beskrivs ska alltid fungera korrekt.

• Koden ska kunna hantera att användaren matar in felaktig input/nonsens utan att gå sönder. Rimlig input-sanitization ska finnas.