Sistemas Operacionais

Aula 10: Verificação

- Teste
- Provadores de teoremas
- Prova totalmente manual
    - Por invariantes
- Verificação automática

Verificação de Programas Paralelos

Extremamente difícil:

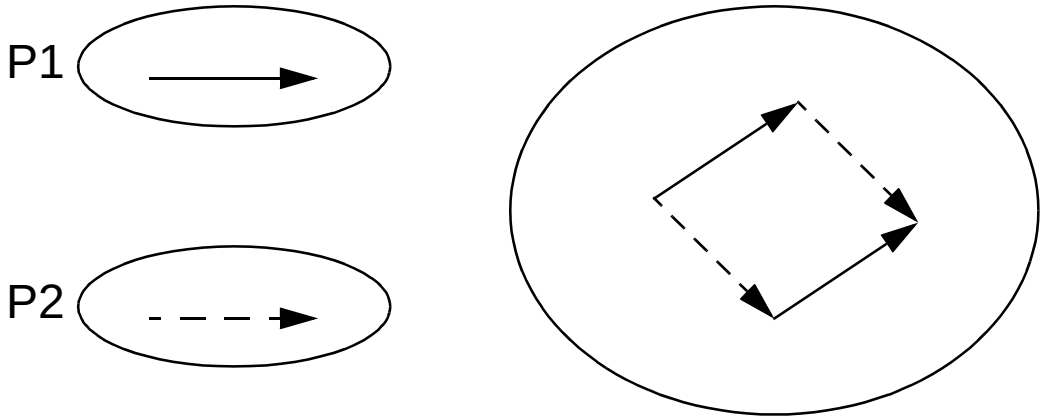
- Tem que levar em conta **toda** a interação entre os processos;
- Manualmente: fácil de esquecer algo;
- Automaticamente: número de estados é muito grande:
  - O número de estados em que o programa pode estar é exponencial no número de processos.

Extremamente importante:

- Virtualmente todos os computadores usam processos concorrentes;
- Inclusive aqueles que controlam sua conta no banco, o avião no qual voce viaja...

Porque tão difícil ?

Composição paralela de processos é cara:



Se P1 tem m estados e P2 tem n estados, P1 X P2 tem m * n estados!

Explosão de estados.

Teste

Implementa-se o programa e executa-se o mesmo *ad nauseam.*

Vantagem: simples, eficiente

Desvantagem: cobertura de falhas muito baixa.

Verificação Formal

Prova-se que o programa está correto:

Métodos matemáticos que derivam propriedades do sistema baseado em assunções e teoremas.

- Teoria complexa;
- Prática nem sempre, as vezes simples;
- Poderosos, mas frequentemente difíceis de usar.

Necessários se você quer confiar no seu programa!

Provadores de Teoremas

Provadores de teoremas são programas que dado:

- Um conjunto de condições satisfeitas pelo sistema, e.g. comunicação é confiável, evento X ocorre ao máximo 10 vezes por segundo, variável Y tem valores entre 0 e 100.

- Um conjunto de regras (teoremas) que permitem estudar o sistema, e.g. Se um valor é enviado de P1 para P2 então fatalmente P2 saberá este valor no futuro.

- Deriva propriedades do sistema, e.g. exclusão mútua não será violada.

Provadores de Teoremas

- Poderosos!
- Difíceis de usar: necessitam intervenção do usuário com frequência, porque o provador se perde!
- Hoje em dia são pouco práticos. Existe muita pesquisa para simplificar sua utilização

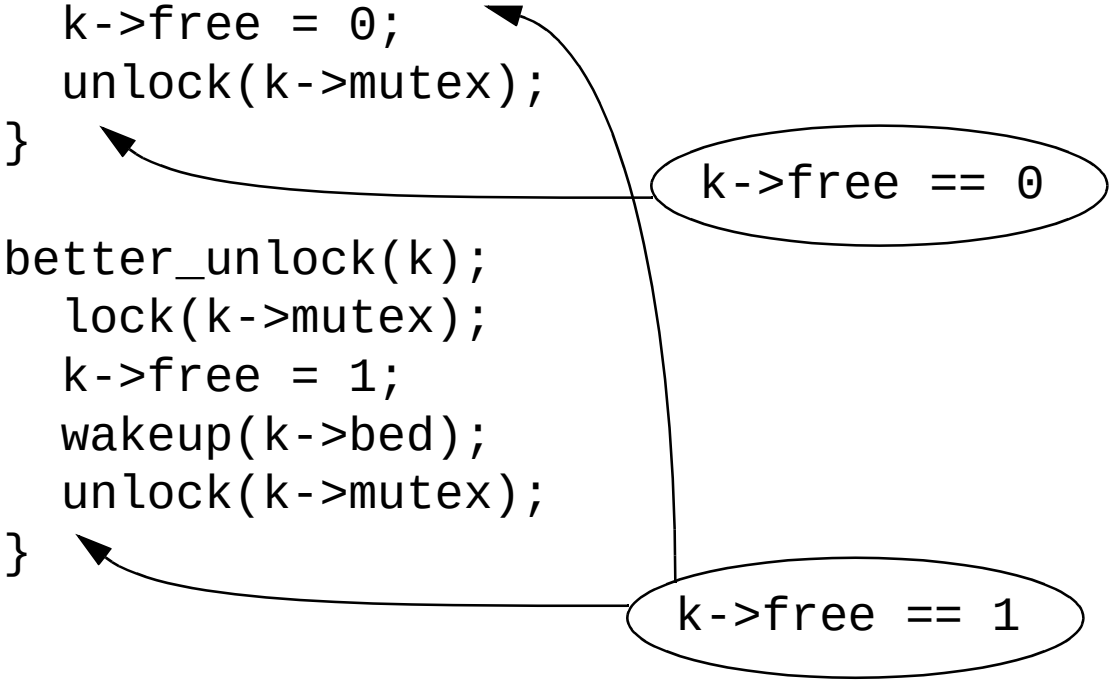
Prova Manual por Invariantes

- Escolhe-se pontos chaves do programa;
- Identificam-se propriedades que devem ser satisfeitas naquele ponto do programa.
- Prova-se que tais propriedades são satisfeitas pelo programa
- Prova-se que a corretude do programa é uma consequência lógica da validade dos invariantes.

Prova Manual por Invariantes

```
better_lock(k) {
  lock(k->mutex);
  if (!k->free)
    sleep(k->bed, k->mutex);
  k->free = 0;
  unlock(k->mutex);
}

better_unlock(k);
  lock(k->mutex);
  k->free = 1;
  wakeup(k->bed);
  unlock(k->mutex);
}
```



Prova Manual por Invariantes

- Uma arte!
- Difícil, mas dá prá fazer.
- Método muito usado.