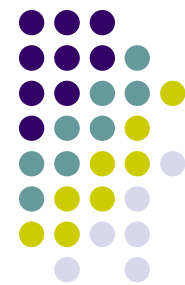




Дебаговање



Дебаговање?

- Буба (енгл. Bug /баг/) - Проблем у програму
- Отклањање буба, проблема
- У ужем смислу то подразумева: контролисано извршавање програма и увид у стање програма зарад отклањања проблема.
- У ширем смислу то су сви могући (и немогући) поступци за откривање и отклањање проблема.



- Програм најчешће ради савршено све док га не покренемо.
- Провера исправности: верификација и валидација
- Верификација проверава да ли програм ради то што ми мислимо да треба да ради.
- Валидација проверава да ли програм ради то што нама треба. (То што ми мислимо да програм треба да ради не мора бити то што нама треба.)
- Тестирање не мора бити формализовано! Али је то више него пожељно код озбиљних подухвата.



Отклањање проблема

- Програм најчешће не ради добро све док га не издебагујемо.
- Испитни случај: улаз (побуда) - очекиван излаз (реакција)
- Зашто добијамо неочекиван излаз?
- Потребан нам је бољи увид у извршавање програма.
- Уобичајено доступне ствари:
 - Контролисано извршавање:
 - Корак по корак и тачке прекида
 - Увид у стање програма
 - Посматрање променљивих и меморије
 - Додавање контролних исписа у код



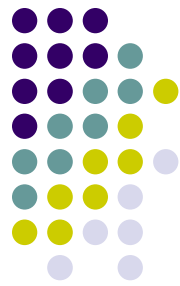
- Неуобичајено доступне ствари:
 - Контролисано извршавање:
 - Корак у назад
 - Условне тачке прекида
 - Кретање по току података, уместо по току извршавања
 - Увид у стање програма
 - Срачунавање израза у тренутном контексту
- Али неки пут немамо ништа од поменутог!
 - Тада се сналазимо како знамо и умемо.



Неке поделе багова

- Ухватљиви (доследни, поновљиви) багови
За одређену побуду увек добијамо исту нежељену реакцију
 - Лако поновљиви багови
 - Побуду која узрокује нежељену реакцију је лако направити у контролисаним условима
 - Тешко поновљиви багови
 - Побуду која узрокује нежељену реакцију је тешко направити у контролисаним условима
- Неухватљиви (недоследни, стохастички) багови
За одређену побуду нежељену реакцију добијамо само понекад
Узрок је постојање нежељеног утицаја на систем ван чинилаца побуде
Могући узроци:
 - Читање из неиницијализоване меморије
 - Ослањање на адресе динамички заузете меморије
 - Упадање у стања недефинисана програмским језиком

NIT Дигресија: недефинисана стања и понашања



- То значи да понашање може бити различито приликом извршавања истог кода на истом систему.

Неки важнији примери недефинисаних понашања у програмском језику Це:

- Прекорачење код интеџерске аритметике
 - Укључујући померање у десно, тзв. шифтовање
- Битске операције над интеџерима
 - Бинарна репрезентација интеџера не мора бити комплемент двојке!
- Важи само за означене интеџере!
 - Неозначени се дефинишу тотално другачије.



Неке поделе багова

- Прости багови

Манифестација проблема је тесно везана за место где постоји грешка у коду.

- Сложени багови

Манифестација проблема и место грешке у коду су врло посредно повезане.



Како успешно дебаговати

- Разумети систем
- Репродуковати баг
- Не претпостављати - погледати
- Подели па владај
- Постепено мењати
- Водити белешке
- Проверити кабел (да није до физичке архитектуре?)
- Применити нов поглед на ствари
- Ништа се не поправља само од себе

Најважнији савет!



**Покушати решити
проблем**