**Multithreading**

1. **Ta'rifi**:
   * Multithreading bitta jarayon (process) ichida bir nechta oqimlarni (thread) ishga tushirishni anglatadi.
   * Oqimlar bir xil xotira maydonini bo'lishadi, bu esa oqimlar orasida kontekstni almashishni jarayonlar orasidagiga qaraganda tezroq qiladi.
2. **Qo'llanish sohalari**:
   * I/O-ga bog'liq vazifalar uchun ideal (masalan, fayllardan o'qish, tarmoq operatsiyalari), bu erda dastur ko'p vaqtni tashqi resurslarni kutishda o'tkazadi.
   * Tez-tez ma'lumot almashish talab qilinadigan vazifalar uchun mos keladi, chunki oqimlar umumiy xotiraga ega bo'ladi.
3. **Afzalliklari**:
   * Xotira samaradorligi yuqori, chunki barcha oqimlar bitta xotira maydonini ishlatadi.
   * Oqimlar orasidagi kontekst almashinuvi jarayonlarga qaraganda tezroq.
4. **Kamchiliklari**:
   * Global Interpreter Lock (GIL) kabi cheklovlar tufayli haqiqiy parallel hisoblashga erishish qiyin bo'lishi mumkin (masalan, Python'da).
   * Xotira almashish tufayli oqimlar orasida ma'lumotlar poygasi (data race) va sinxronizatsiya muammolari paydo bo'lishi mumkin.

**Multiprocessing**

1. **Ta'rifi**:
   * Multiprocessing bir nechta jarayonlarni ishga tushirishni anglatadi, har bir jarayon o'z xotira maydoniga ega.
   * Har bir jarayon operatsion tizim tomonidan alohida boshqariladi.
2. **Qo'llanish sohalari**:
   * CPU-ga bog'liq vazifalar uchun ideal (masalan, og'ir hisoblash vazifalari) bo'lib, bu yerda har bir jarayon alohida yadrolarda parallel ravishda ishlay oladi.
   * Katta miqdordagi ma'lumotlarni mustaqil ravishda qayta ishlash kerak bo'lgan vazifalar uchun mos.
3. **Afzalliklari**:
   * Haqiqiy parallel hisoblashni amalga oshirishi mumkin, chunki har bir jarayon o'zining alohida xotira maydoniga ega va alohida yadrolarda ishlashi mumkin.
   * Global Interpreter Lock (GIL) cheklovlari mavjud emas, shuning uchun to'liq CPU quvvatidan foydalanish mumkin.
4. **Kamchiliklari**:
   * Xotira foydalanishi ko'proq bo'lishi mumkin, chunki har bir jarayon o'z xotira maydoniga ega.
   * Jarayonlar orasidagi kontekst almashinuvi oqimlarga qaraganda sekinroq.
   * Ma'lumotlarni jarayonlar orasida almashish qiyinroq va ko'proq kod yozishni talab qilishi mumkin.

**Xulosa**

* **Multithreading**: I/Oga bog'liq vazifalar va umumiy xotiradan foydalanadigan tez-tez ma'lumot almashinuvi talab qilinadigan vazifalar uchun mos.
* **Multiprocessing**: CPUga bog'liq og'ir hisoblash vazifalari va alohida-alohida qayta ishlanishi kerak bo'lgan katta miqdordagi ma'lumotlar uchun mos.