

اولی

\neq
wait(s1)
wait(s2)

\neq
wait(s1)
wait(s2)

//cs
signal(s1)
signal(s2)

//cs
signal(s1)
signal(s2)

در اولی circular wait
در هر پروسسی باید فضای درختی وجود داشته باشد
increasing order of enumeration

دسته بندی
توی مثال راه حل به جا کردن s1 و s2 در P1 و P2 نیست ایجاب می کنه
صورت order به اجزای سیستم فرها راجع.

در دومی no preemption
اگر یک پروسسی یک ریسورس گرفته بود و منتظر ریسورس دیگری
بود و ریسورس دیگر را کسی نمی رفع و ریسورس های قبلی را هم از پی می گیریم. وقتی اتمام می کنه

P1
P2
wait(s1)
wait(s2)
wait(s1, s2)
اگر s2 را نتوانست بگیره و s1 را هم
آزاد کنه اگر نه با هم در امانه دره

P1
P2
wait(s2)
wait(s1, s2)
اگر s1 را نتوانست بگیره
s2 را هم آزاد کنه
در شبیه صورت همدار داره
signal(s2)
signal(s1)

signal(s1)
~ (s2)