

آزمایشگاه سیستم‌های عامل

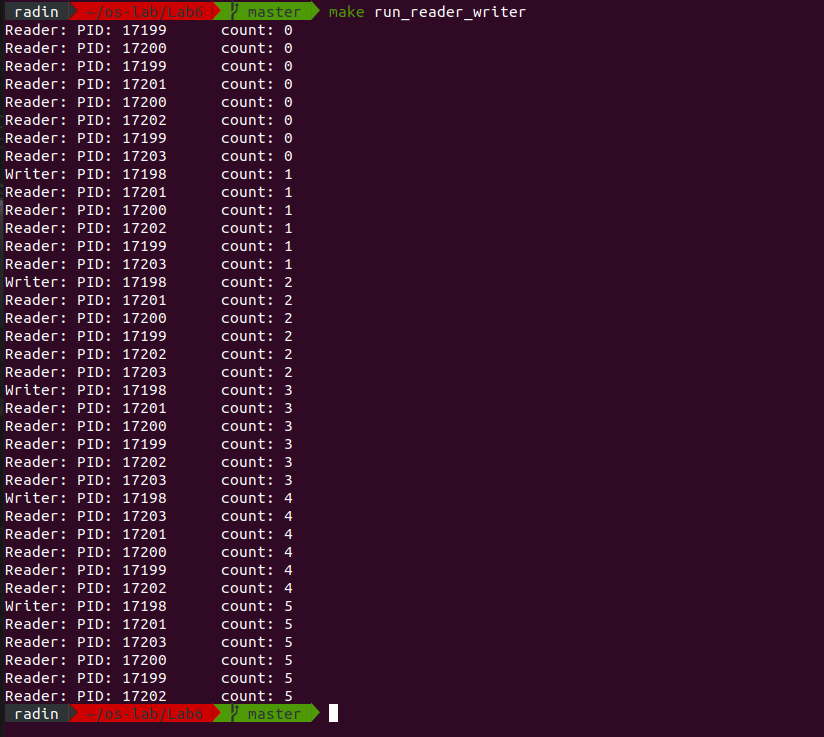
رادین شایانفر

پاییز ۱۳۹۹

* **مسئله خوانندگان-نویسندگان:**

با دستور زیر برنامه را اجرا می‌کنیم:

make run\_reader\_writer

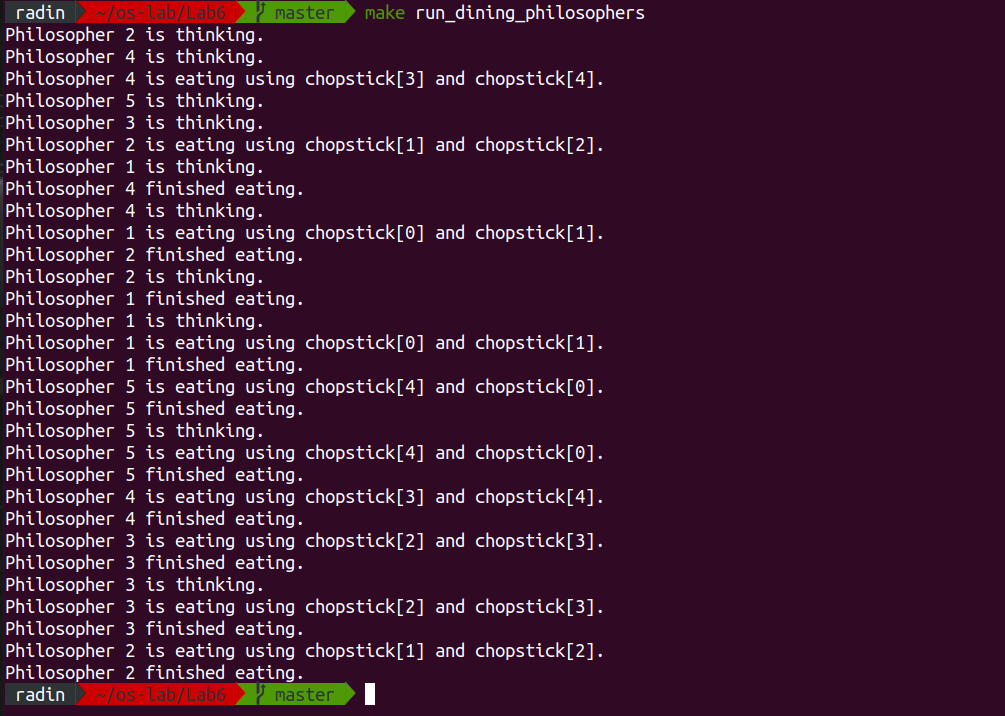
****در این برنامه ۵ خواننده و یک نویسنده به طور مشترک به متغیر count دسترسی دارند. به دلیل وجود شرایط مسابقه برای جلوگیری از مشکل از ۲ قفل mutex و rw\_mutex به صورت مشترک بین فرایندها استفاده شده است. در این برنامه اولین خواننده‌ای که وارد می‌شود قفل rw\_mutex را می‌گیرد و آخرین خواننده‌ای که خارج می‌شود آن را آزاد می‌کند. برای شمارش تعداد خواننده‌های وارد شده نیز از متغیر read\_count استفاده شده است که با قفل mutex انحصار متقابل آن تضمین می‌شود.

**شکل (۱) – اجرای کد مسئله‌ی خوانندگان-نویسندگان**

* **مسئله فیلسوف‌های غذاخور:**

با دستور زیر برنامه را اجرا می‌کنیم:

make run\_dining\_philosophers

در این مسئله از ۵ قفل chopstick[5] برای جلوگیری از ورود هم‌زمان به ناحیه بحرانی استفاده شده است. هر فیلسوف در ترد جداگانه‌ای به تعداد بار مشخص (به طور خاص ۲ بار در شکل زیر) ابتدا فکر می‌کند، سپس ۱ ثانیه غذا می‌خورد. هر چند این روش انحصار متقابل را تضمین می‌کند اما امکان به وجود آمدن بن‌بست (در صورتی که همه‌ی فیلسوف‌ها ابتدا قفل اول خود را، پیش از این که کسی قفل دوم خود را در اختیار بگیرد، در اختیار بگیرند) در آن وجود دارد.

**شکل (۲) – اجرای کد مسئله‌ی فیلسوف‌های غذاخور**