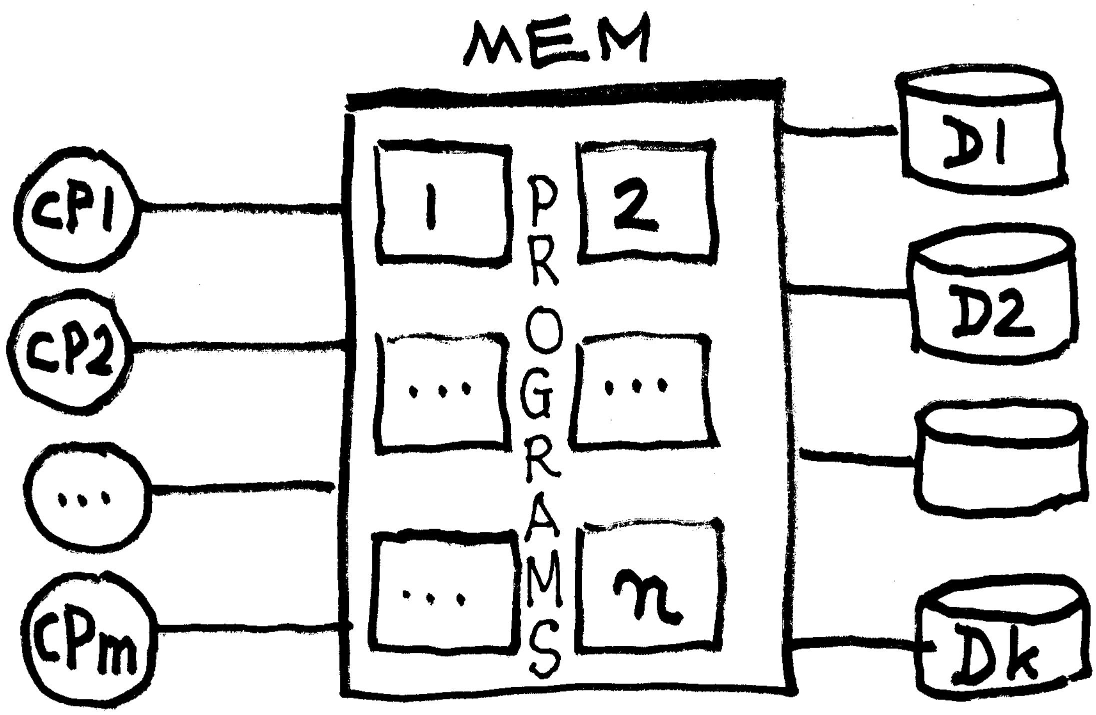
MULTIPROGRAMMING WITH n IDENTICAL JOBS

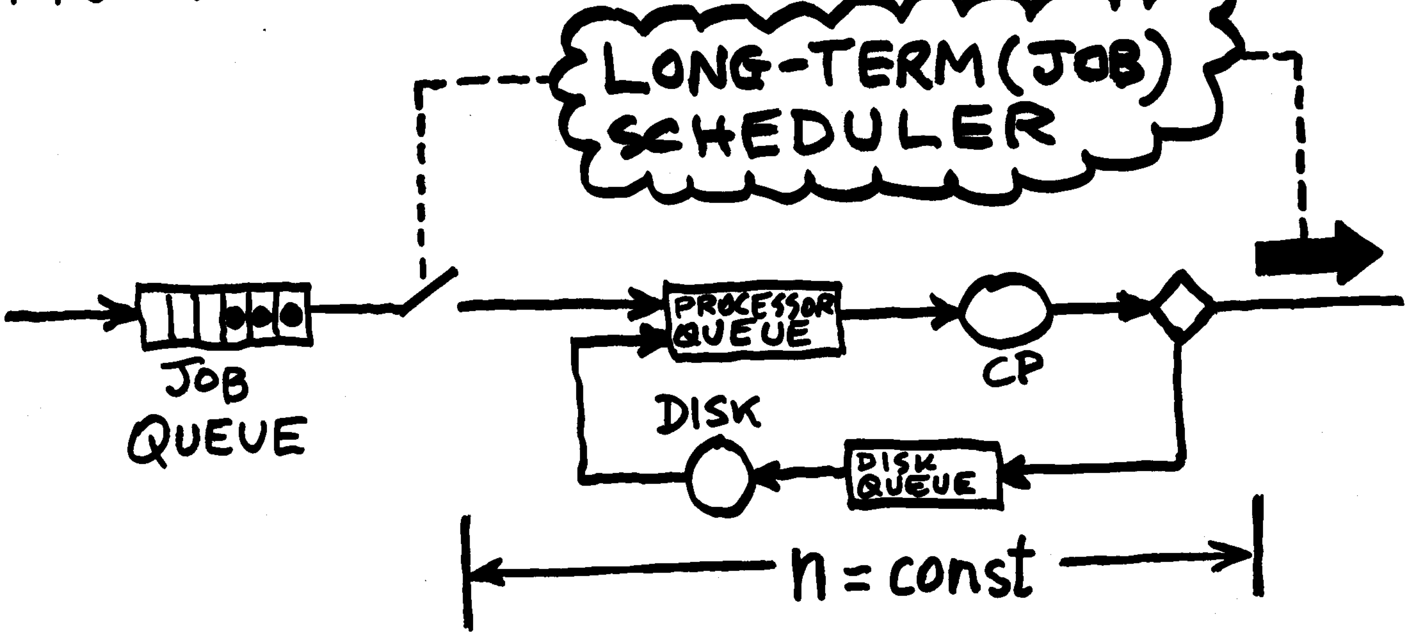
Parameters:

k = number of disks
m = number of processors
n = number of programs



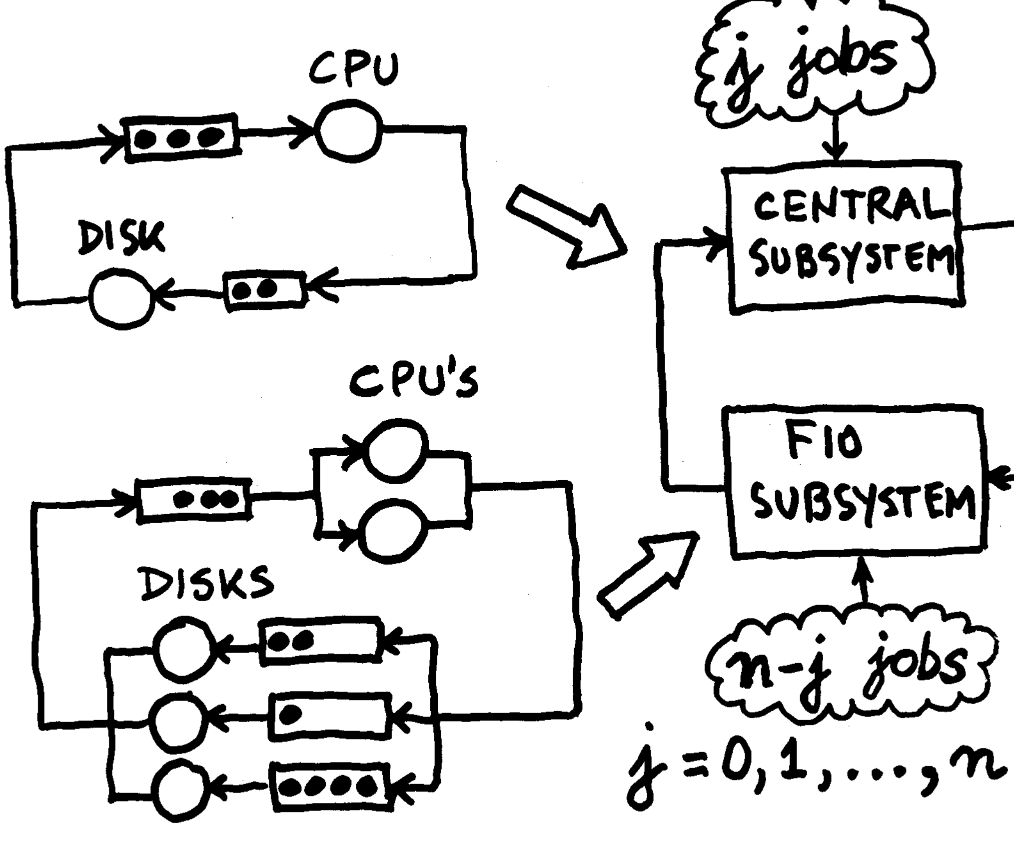
Dynamic model = closed queuing network

STABLE (CONSTANT) DEGREE OF MULTIPROGRAMMING

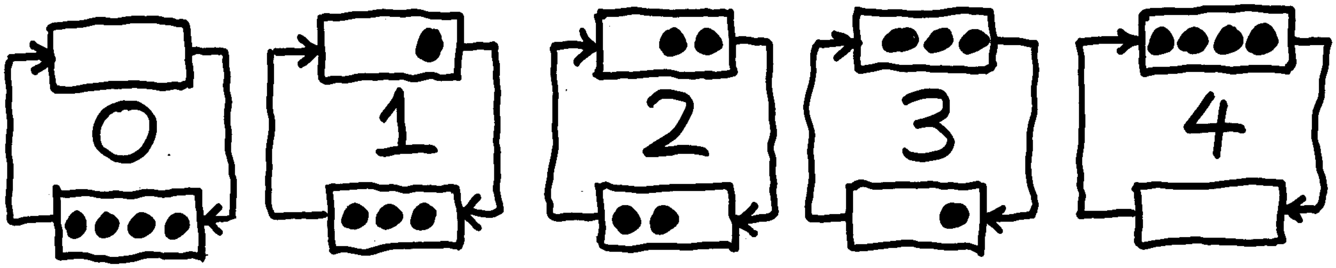


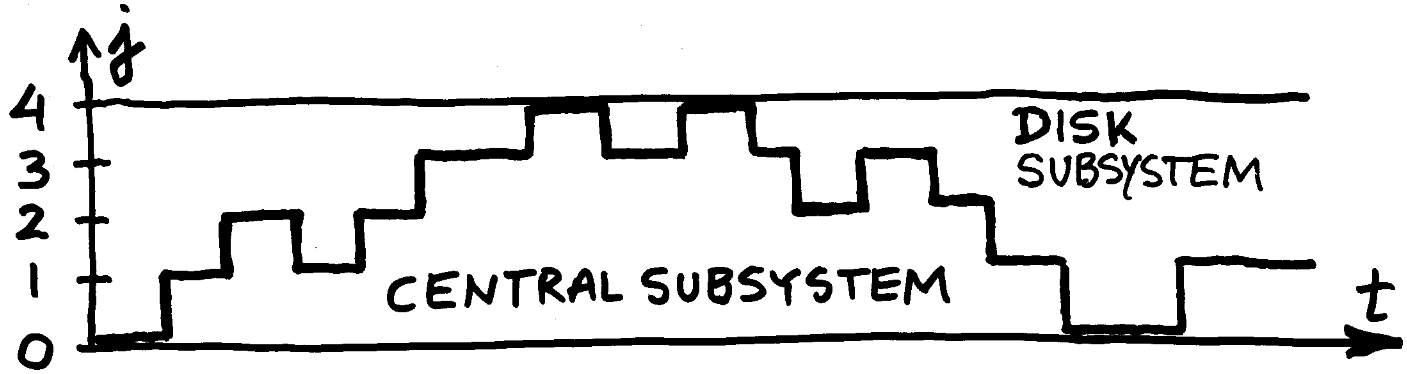
LONG-TERM SCHEDULER CONTROLS THE DEGREE OF MULTIPROGRAMMING

DYNAMIC MODELS: CLOSED QUEUEING NETWORKS

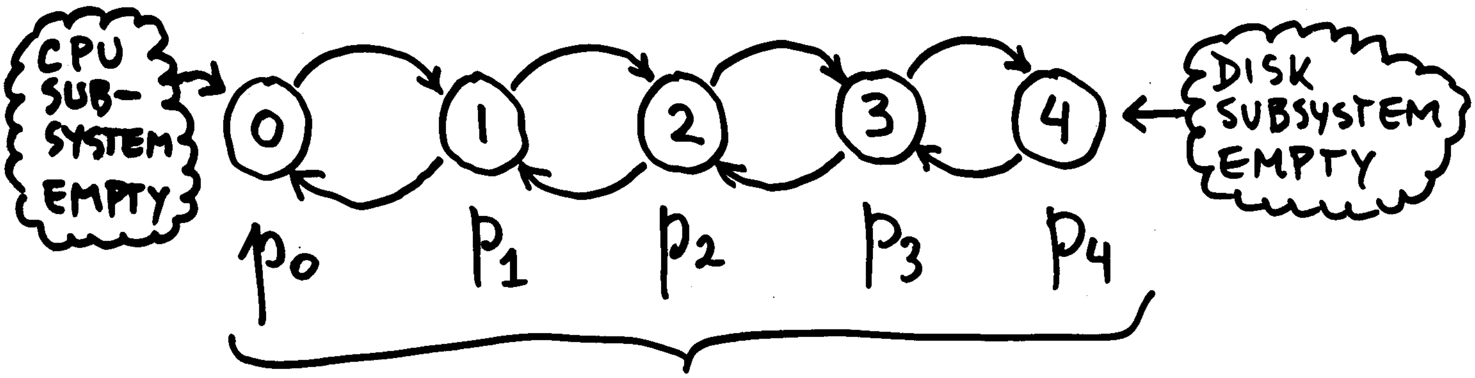


DISCRETE STATES





STATE TRANSITION DIAGRAM



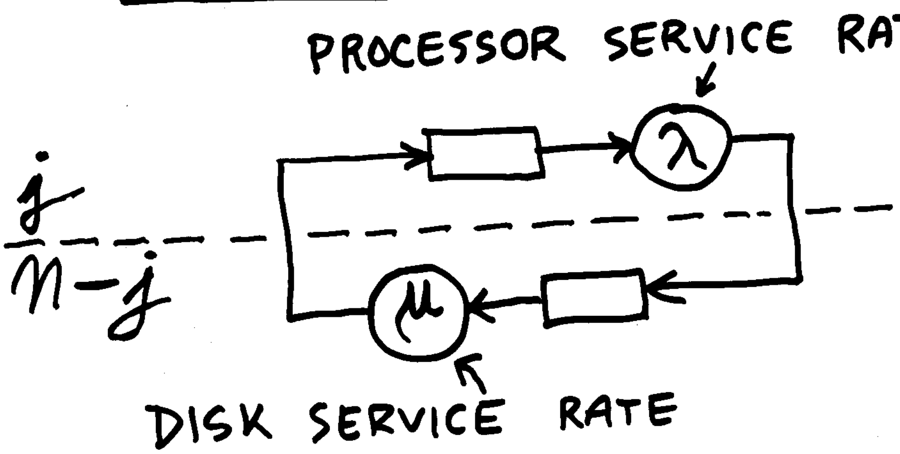
Probabilities of states: $p_0+p_1+p_2+p_3+p_4=1$

$p_0 = Pr$ [all processors are idle]

$p_4 = Pr$ [all disks are idle]

Single processor systems: $U_p = 1 - p_0$

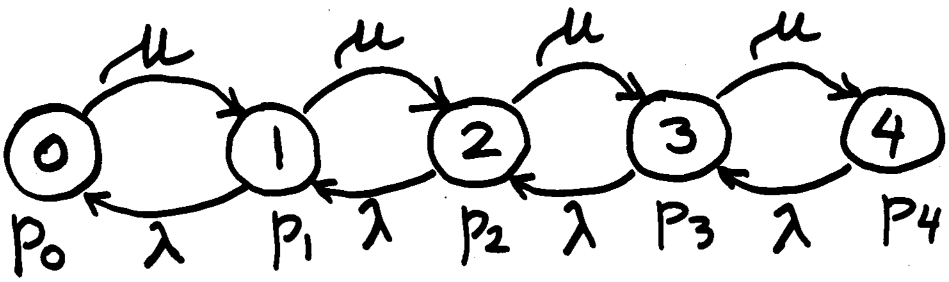
BALANCE EQUATIONS



$\lambda = 1/S_p$

$\mu = 1/S_d$

$n = 4$

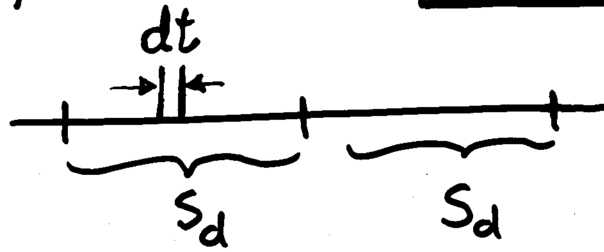


$$\boxed{p_i \mu = p_{i+1} \lambda}, \quad i = 0, 1, 2, 3$$

WHY ?

$$p_i \mu \, dt = p_{i+1} \lambda \, dt \qquad dt \to 0$$

$$p_i \mu \, dt = p_i \frac{dt}{S_d}$$



= Pr[state i] $\cdot$ Pr [transition $i \to i+1$ during interval dt]

= Pr [state $i+1$] $\cdot$ Pr [transition $i+1 \to i$ during interval dt]

$\therefore$ Transitions $i \to i+1$ must be balanced with transitions $i+1 \to i$

$$p_{i+1} = \frac{\mu}{\lambda} p_i = \frac{s_p}{s_d} p_i = \rho p_i , \quad \rho = \frac{s_p}{s_d}$$

$$\boxed{\begin{aligned} p_1 &= \rho p_0 \\ p_2 &= \rho p_1 = \rho^2 p_0 \\ &\vdots \\ p_n &= \rho p_{n-1} = \rho^n p_0 \\ p_0 + p_1 &+ \cdots + p_n = 1 \end{aligned}}$$

$$p_0 + \rho p_0 + \rho^2 p_0 + \cdots + \rho^n p_0 = 1$$

$$p_0 (1 + \rho + \rho^2 + \cdots + \rho^n) = 1$$

$$p_0 \frac{\rho^{n+1} - 1}{\rho - 1} = 1$$

$$\boxed{\begin{aligned} p_0 &= \frac{\rho - 1}{\rho^{n+1} - 1} , \quad U_p = 1 - p_0 = \frac{\rho^{n+1} - \rho}{\rho^{n+1} - 1} \\ p_i &= \frac{\rho^i (\rho - 1)}{\rho^{n+1} - 1} \end{aligned}}$$

Computation of sum $1+x+x^2+\dots+x^n$

$$(1+x+x^2+\dots+x^n)\cdot(1-x) =$$

$$\begin{aligned} = 1 + x + x^2 + \dots + x^n & \\ - x - x^2 - \dots - x^n & - x^{n+1} \end{aligned}$$

$$= 1 - x^{n+1}$$

$$\therefore \quad 1+x+x^2+\dots+x^n = \frac{1-x^{n+1}}{1-x} = \frac{x^{n+1}-1}{x-1}$$

Special case $|x|<1$, $n \to +\infty$

$$1+x+x^2+\dots = \frac{1}{1-x}$$

$$S = 1+x+x^2+\dots = 1 + x(1+x+x^2+\dots)$$

$$= 1 + xS$$

$$S = \frac{1}{1-x}$$

$$\lim_{n\to\infty} U_p = \lim_{n\to\infty} \frac{\rho^{n+1}-\rho}{\rho^{n+1}-1} = \begin{cases} \rho\ , & \rho < 1 \\ & (S_p < S_d) \\ 1\ , & \rho > 1 \end{cases}$$

$$U_p\Big|_{n=1} = \frac{\rho^2-\rho}{\rho^2-1} = \frac{\rho(\rho-1)}{(\rho-1)(\rho+1)} = \frac{\rho}{\rho+1}$$

<u>Example:</u>

$S_p = 2\,ms$

$S_d = 10\,ms$

$\rho = S_p / S_d = 0.2$

$U_{p\,min} = \frac{0.2}{1.2} = 0.167$

$U_{p\,max} = \rho = 0.2$

$$p_n = \frac{\rho^n(\rho-1)}{\rho^{n+1}-1} = \frac{\rho^{n+1}-\rho^n}{\rho^{n+1}-1}$$

$$U_d = 1 - p_n = \frac{\rho^n-1}{\rho^{n+1}-1} = \begin{cases} \frac{1}{\rho+1}\ , & n=1 \\ 1\ , & n\to\infty \end{cases}$$

$U_{d\,min} = \frac{1}{1.2} = 0.83 \qquad U_{d\,max} = 1$

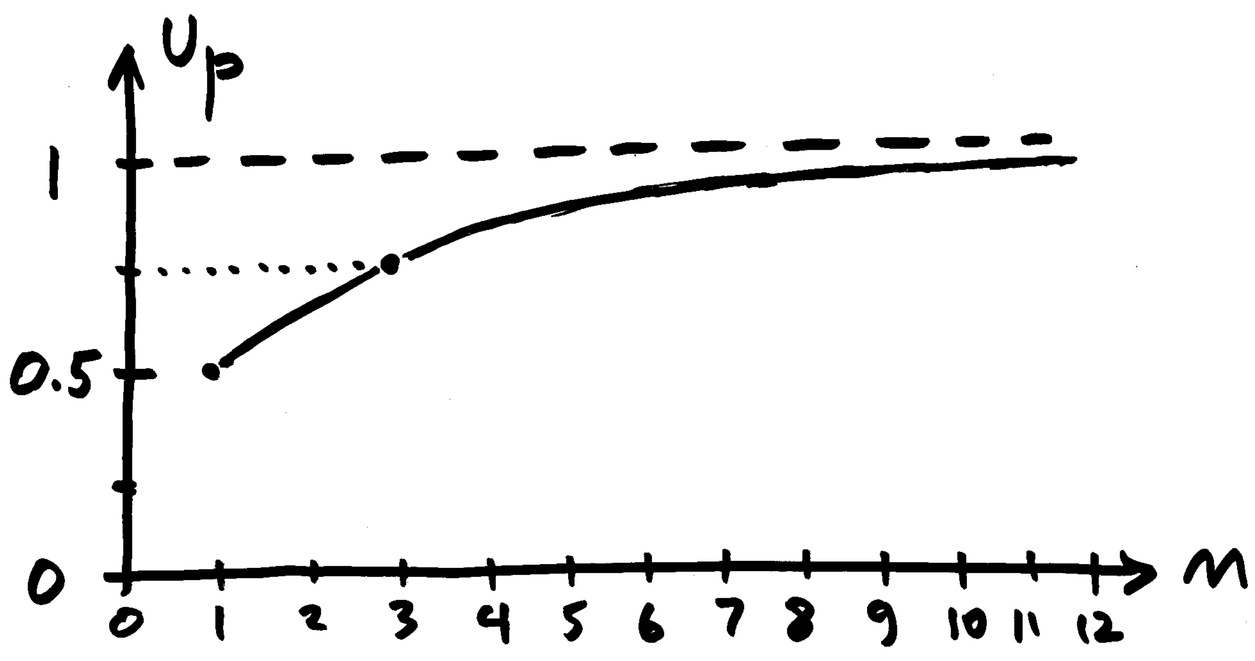
The case $S_p = S_d$

$\rho = 1$

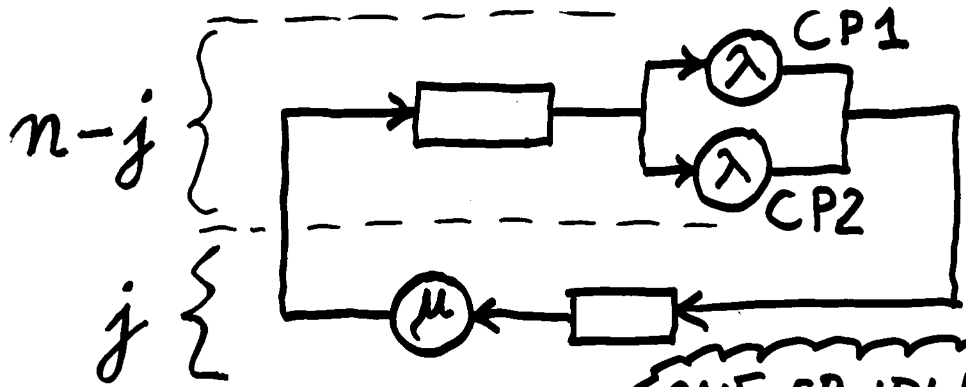
$$U_p = \frac{\rho^{n+1} - \rho}{\rho^{n+1} - 1}$$

$$\lim_{\rho \to 1} \frac{\rho^{n+1} - \rho}{\rho^{n+1} - 1} = \lim_{\rho \to 1} \frac{(n+1)\rho^{n} - 1}{(n+1)\rho^{n}} = \frac{n}{n+1}$$

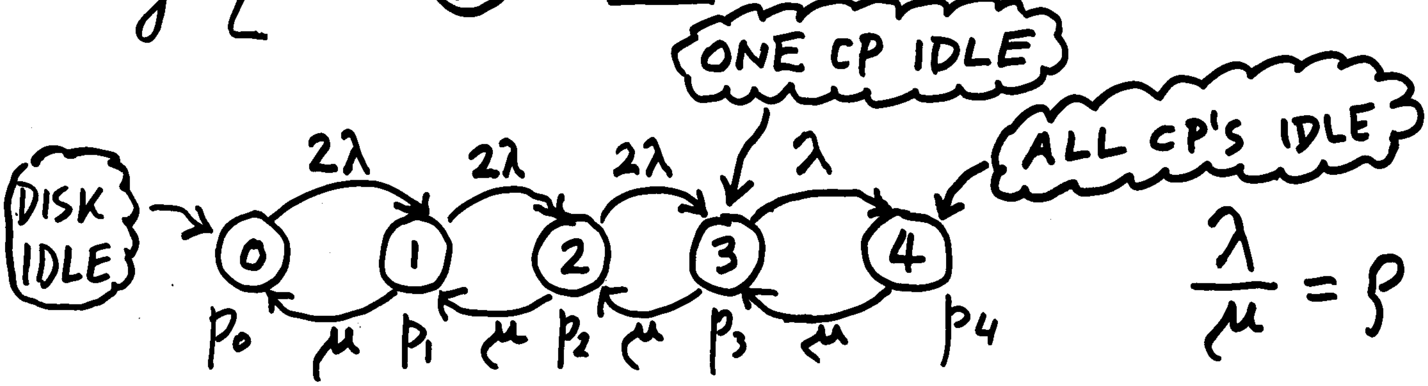
$$U_p \Big|_{S_p = S_d} = \frac{n}{n+1}$$



THE CASE OF TWO PROCESSORS



$n = 4$



$$\frac{\lambda}{\mu} = \rho$$

$$2\lambda p_0 = \mu p_1 \quad \rightarrow \quad p_1 = 2\rho p_0$$

$$2\lambda p_1 = \mu p_2 \quad \rightarrow \quad p_2 = 2\rho p_1 = 4\rho^2 p_0$$

$$2\lambda p_2 = \mu p_3 \quad \rightarrow \quad p_3 = 2\rho p_2 = 8\rho^3 p_0$$

$$\lambda p_3 = \mu p_4 \quad \rightarrow \quad p_4 = \rho p_3 = 8\rho^4 p_0$$

$$p_0 + p_1 + p_2 + p_3 + p_4 = 1$$

$$p_0 (1 + 2\rho + 4\rho^2 + 8\rho^3 + 8\rho^4) = 1$$

$$p_0 = \frac{1}{1 + 2\rho + 4\rho^2 + 8\rho^3 + 8\rho^4}$$

$$U_p = p_0 + p_1 + p_2 + \frac{1}{2} p_3 = \frac{1 + 2\rho + 4\rho^2 + 4\rho^3}{1 + 2\rho + 4\rho^2 + 8\rho^3 + 8\rho^4}$$

$$U_d = 1 - p_0$$

T_p = total processor time per job

R = response time (4 jobs, same priority level)

RU_p = processor time delivered by a single processor

$2RU_p$ = total processor time delivered by two processors

$4T_p$ = total processor time required by 4 jobs

$$2RU_p = 4T_p$$

$$R = 2T_p/U_p = \frac{2T_p(1+2\rho+4\rho^2+8\rho^3+8\rho^4)}{1+2\rho+4\rho^2+4\rho^3}$$

Generally:

$$m\,RU_p = n\,T_p \rightarrow R = \frac{nT_p}{mU_p}$$

n: Degree of multiprogramming

m: Number of processors