

Comenzado el	viernes, 1 de octubre de 2021, 19:00
Estado	Finalizado
Finalizado en	viernes, 1 de octubre de 2021, 19:23
Tiempo empleado	23 minutos 24 segundos
Calificación	9,25 de 10,00 (93%)

1

Puntúa 2,00 sobre 2,00

Correcta

Tots els sistemes operatius disposen d'un **planificador de processos** encarregat de repartir l'ús del processador de la forma més eficient possible i assegurant que tots els processos s'executin en algun moment. Per realitzar la planificació, el sistema operatiu es basarà en l'estat dels processos per saber quins necessitaran l'ús del processador. Els processos en disposició de ser executats s'organitzaran en una cua esperant el seu torn.

Ompliu els buits per representar que els estats més habituals tinguin en compte que tot comença amb un **nou** procés. El primer estat que ens trobem és **nou**, és a dir, quan un procés és creat. Un cop creat, passa a l'estat de . En aquest moment el procés està llest per fer ús del processador, està competint pel recurs del processador. El planificador de processos del sistema operatiu és el que decideix quan entra el procés a executar-se. Quan el procés s'està executat, el seu estat s'anomena . Un altre cop, és el planificador l'encarregat de decidir quan abandona el processador.

Des de l'estat d'execució, un procés pot passar a l'estat de . En aquest estat, el procés estarà a l'espera d'un esdeveniment, com pot ser una operació d'entrada/sortida, o l'espera de la finalització d'un altre procés, etc. Quan l'esdeveniment esperat succeeixi, el procés tornarà a l'estat de preparat.

L'últim estat és . És un estat al qual s'arriba un cop el procés ha finalitzat tota la seva execució i estarà a punt per tal que el sistema n'alliberi quan pugui els recursos associats.

La teva resposta és correcta.

La respuesta correcta es:

Tots els sistemes operatius disposen d'un **planificador de processos** encarregat de repartir l'ús del processador de la forma més eficient possible i assegurant que tots els processos s'executin en algun moment. Per realitzar la planificació, el sistema operatiu es basarà en l'estat dels processos per saber quins necessitaran l'ús del processador. Els processos en disposició de ser executats s'organitzaran en una cua esperant el seu torn.

Ompliu els buits per representar que els estats més habituals tinguin en compte que tot comença amb un nou procés.

El primer estat que ens trobem és **nou**, és a dir, quan un procés és creat. Un cop creat, passa a l'estat de **[Preparat]**. En aquest moment el procés està llest per fer ús del processador, està competint pel recurs del processador. El planificador de processos del sistema operatiu és el que decideix quan entra el procés a executar-se. Quan el procés s'està executat, el seu estat s'anomena **[en execució]**. Un altre cop, és el planificador l'encarregat de decidir quan abandona el processador.

Des de l'estat d'execució, un procés pot passar a l'estat de **[bloquejat]**. En aquest estat, el procés estarà a l'espera d'un esdeveniment, com pot ser una operació d'entrada/sortida, o l'espera de la finalització d'un altre procés, etc. Quan l'esdeveniment esperat succeeixi, el procés tornarà a l'estat de preparat.

L'últim estat és **[acabat]**. És un estat al qual s'arriba un cop el procés ha finalitzat tota la seva execució i estarà a punt per tal que el sistema n'alliberi quan pugui els recursos associats.

2

Puntúa 0,75 sobre 0,75

Correcta

Quina és la definició de L'atomicitat d'una operació

Seleccione una:

- ☐ a. és poder garantir que no es poden executar dues operacions de forma concurrent si fan ús d'un recurs compartit fins que una de les dues **comparteix** aquest recurs. Una operació atòmica únicament ha d'observar dos estats: l'inicial i el resultat. Una operació atòmica, o s'executa completament o no ho fa en absolut.
- ☐ b. és poder garantir que es poden executar dues operacions de forma concurrent si fan ús d'un recurs compartit . Una operació atòmica únicament ha d'observar dos estats: l'inicial i el resultat. Una operació atòmica, o **s'executa completament** o no ho fa en absolut.
- ☐ c. és poder garantir que no es poden executar dues operacions de forma concurrent si fan ús d'un recurs compartit fins que una de les dues deixa **lliure aquest** recurs. Una operació atòmica únicament ha d'observar dos estats: l'inicial i el resultat. Una operació atòmica, pot quedar-se a mig executar sempre i quan realitzem un ROLLBACK.
- ☒ d. és poder garantir que no es poden executar dues operacions de forma concurrent si fan ús d'un recurs compartit fins que una de les dues **deixa lliure** aquest recurs. Una operació atòmica únicament ha d'observar dos estats: l'inicial i el resultat. Una operació atòmica, o s'executa completament o no ho fa en absolut. ✓

La resposta és correcta.

La respuesta correcta es: és poder garantir que no es poden executar dues operacions de forma concurrent si fan ús d'un recurs compartit fins que una de les dues **deixa lliure** aquest recurs. Una operació atòmica únicament ha d'observar dos estats: l'inicial i el resultat. Una operació atòmica, o s'executa completament o no ho fa en absolut.

3

Puntúa 0,75 sobre 0,75

Correcta

Ompliu els buits amb les opcions possibles

La classe ✓ és una classe abstracta per les tasques que s'executen a ✓ i conté els mètodes `fork` i `join`. El mètode ✓ situa la tasca invocada a la cua d'execucions en qualsevol moment per tal que sigui planificada. Això permet a una tasca crear-ne de noves i deixar-les a punt per ser executades quan el gestor ho consideri.

El mètode ✓ aturarà l'execució del fil invocador a l'espera que la tasca invocada finalitzi l'execució i retorni si fos el cas els resultats. El bloqueig del fil posarà en alerta el gestor `ForkJoinPool` que podrà intercanviar la tasca aturada per una altra que resti en espera.

La resposta és correcta.

La respuesta correcta es:

Ompliu els buits amb les opcions possibles

La classe `[ForkJoinTask]` és una classe abstracta per les tasques que s'executen a `[ForkJoinPool]` i conté els mètodes `fork` i `join`. El mètode `[fork()]` situa la tasca invocada a la cua d'execucions en qualsevol moment per tal que sigui planificada. Això permet a una tasca crear-ne de noves i deixar-les a punt per ser executades quan el gestor ho consideri.

El mètode `[join()]` aturarà l'execució del fil invocador a l'espera que la tasca invocada finalitzi l'execució i retorni si fos el cas els resultats. El bloqueig del fil posarà en alerta el gestor `ForkJoinPool` que podrà intercanviar la tasca aturada per una altra que resti en espera.

4

Puntúa 0,75 sobre 0,75

Correcta

Responen si és vertadera (V) o falsa (F) la següent afirmació:

Quan dos fils s'estan executant concurrentment i cada un té un bloqueig exclusiu que l'altre necessita per continuar es produeix un interbloqueig.

Seleccione una:

- ☒ Verdadero ✓
- ☐ Falso

La respuesta correcta es 'Verdadero'

5

Puntúa 0,75 sobre 0,75

Correcta

Arrossegueu les diverses opcions que teniu per omplir els buits

- La ✓ és programació que executa tots els seus processos a la vegada.
- Quant la programació concurrent es realitza en un sistema multiprocessador parlem de ✓. L'execució dels processos es reparteix per les diferents CPU.
- La ✓ és un tipus de programació concurrent en la qual els processos són executats en una ✓, és a dir, els processos s'executen en diferents equips.

La resposta és correcta.

La respuesta correcta es:

Arrossegueu les diverses opcions que teniu per omplir els buits

- La [programació concurrent] és programació que executa tots els seus processos a la vegada.
- Quant la programació concurrent es realitza en un sistema multiprocessador parlem de [programació paral·lela]. L'execució dels processos es reparteix per les diferents CPU.
- La [programació distribuïda] és un tipus de programació concurrent en la qual els processos són executats en una [xarxa de sistemes distribuïts], és a dir, els processos s'executen en diferents equips.

Si mirem la diferència que hi ha entre sistemes multiprocessadors fortament acoblats i dèbilment acoblats podríem fer les següents afirmacions. Arrossegueu les opcions que teniu per omplir els buits

En una arquitectura de sistemes [multiprocessadors fortament acoblats] ✓, els diferents processadors comparteixen una mateixa memòria i estan interconnectats, tot i que també poden tenir una petita memòria cau en el mateix processador. Hi ha una forta col·laboració entre processadors, actualment la majoria de [Sistemes Operatius] ✓ suporten aquest tipus de sistemes multiprocessador.

Els [multiprocessadors dèbilment acoblats] ✓ no comparteixen memòria, cada processador té una memòria associada. Un tipus de sistemes poc acoblats són els [sistemes distribuïts] ✓.

La resposta és correcta.

La respuesta correcta es:

Si mirem la diferència que hi ha entre sistemes multiprocessadors fortament acoblats i dèbilment acoblats podríem fer les següents afirmacions. Arrossegueu les opcions que teniu per omplir els buits

En una arquitectura de sistemes [multiprocessadors fortament acoblats], els diferents processadors comparteixen una mateixa memòria i estan interconnectats, tot i que també poden tenir una petita memòria cau en el mateix processador. Hi ha una forta col·laboració entre processadors, actualment la majoria de [Sistemes Operatius] suporten aquest tipus de sistemes multiprocessador.

Els [multiprocessadors dèbilment acoblats] no comparteixen memòria, cada processador té una memòria associada. Un tipus de sistemes poc acoblats són els [sistemes distribuïts].


Runnable és una interfície.






- Seleccione una:
- ☒ Verdadero ✓
 - ☐ Falso



La respuesta correcta es 'Verdadero'


Selecioneu les paraules que falten

Hem definit programació concurrent o concurrència com la tècnica per la qual múltiples processos s’executen alhora i poden comunicar-se entre ells. La majoria dels sistemes intenten aprofitar aquesta concurrència per incrementar la capacitat d’execució.

 és un dels principals avantatges. Quan s’executen diversos processos alhora, la velocitat d’execució global es pot incrementar. Cal tenir en compte, però, que no sempre és així. A vegades, depenent de la complexitat de l’aplicació, les tècniques de sincronisme o comunicació són més costoses en temps que l’execució dels processos.

Un sistema multiprocessador és , ja que, si augmenten els processos que s’estan executant és capaç de  la càrrega de treball dels processadors, i pot també reassignar  els recursos de memòria i els dispositius per ser més eficients. És de . Si el sistema ho permet, es poden afegir nous processadors de forma senzilla i així augmenten la seva .

Els sistemes multiprocessador poden ser . El fet de disposar de diversos processadors, pot permetre un augment de la disponibilitat dels recursos (més processadors al servei de l’usuari) o bé l’ús de processadors especialitzats en tasques de verificació i control. En el darrer cas parlarem de sistemes amb una alta . Una fallada d’un processador no fa que el sistema s’aturi.

Els sistemes multiprocés ens permeten diferenciar processos per la seva  i, per tant, reservar processadors per operacions complexes, aprofitar-ne d’altres per processaments paral·lels i per avançar l’execució.

La resposta és correcta.

La respuesta correcta es:

Selecioneu les paraules que falten

Hem definit programació concurrent o concurrència com la tècnica per la qual múltiples processos s’executen alhora i poden comunicar-se entre ells. La majoria dels sistemes intenten aprofitar aquesta concurrència per incrementar la capacitat d’execució.

[Incrementar la potència de càlcul i el rendiment] és un dels principals avantatges. Quan s’executen diversos processos alhora, la velocitat d’execució global es pot incrementar. Cal tenir en compte, però, que no sempre és així. A vegades, depenent de la complexitat de l’aplicació, les tècniques de sincronisme o comunicació són més costoses en temps que l’execució dels processos.

Un sistema multiprocessador és [flexible], ja que, si augmenten els processos que s’estan executant és capaç de [distribuir] la càrrega de treball dels processadors, i pot també reassignar [dinàmicament] els recursos de memòria i els dispositius per ser més eficients. És de [fàcil creixement]. Si el sistema ho permet, es poden afegir nous processadors de forma senzilla i així augmenten la seva [potència].

Els sistemes multiprocessador poden ser [sistemes redundants]. El fet de disposar de diversos processadors, pot permetre un augment de la disponibilitat dels recursos (més processadors al servei de l’usuari) o bé l’ús de processadors especialitzats en tasques de verificació i control. En el darrer cas parlarem de sistemes amb una alta [tolerància a fallades]. Una fallada d’un processador no fa que el sistema s’aturi.

Els sistemes multiprocés ens permeten diferenciar processos per la seva [especialització] i, per tant, reservar processadors per operacions complexes, aprofitar-ne d’altres per processaments paral·lels i per avançar l’execució.

9

Puntúa 0,75 sobre 0,75

Correcta

work-stealing busca fils poc actius per intercanviar tasques entre ells i millorar el rendiment del sistema.

Selecione una:

- ☒ Verdadero ✓
- ☐ Falso

La respuesta correcta es 'Verdadero'

10

Puntúa 0,00 sobre 0,75

Incorrecta

Digueu si la següent afirmació és certa o falsa.

Un exemple clàssic de programació recursiva és el model de **productor-consumidor**. Es tracta d'un fil (productor) que genera una sortida que és agafada per un altre fil (consumidor).

Imaginem un sistema en el qual un fil (productor) va generant nombres aleatoris i un altre fil (consumidor) els va agafant per mostrar-los.

Selecione una:

- ☒ Verdadero ✗
- ☐ Falso

La respuesta correcta es 'Falso'

◀ ENUNCIAT EAC1P1

Ir a...

M9 LLIURAMENT EAC1P1 ▶