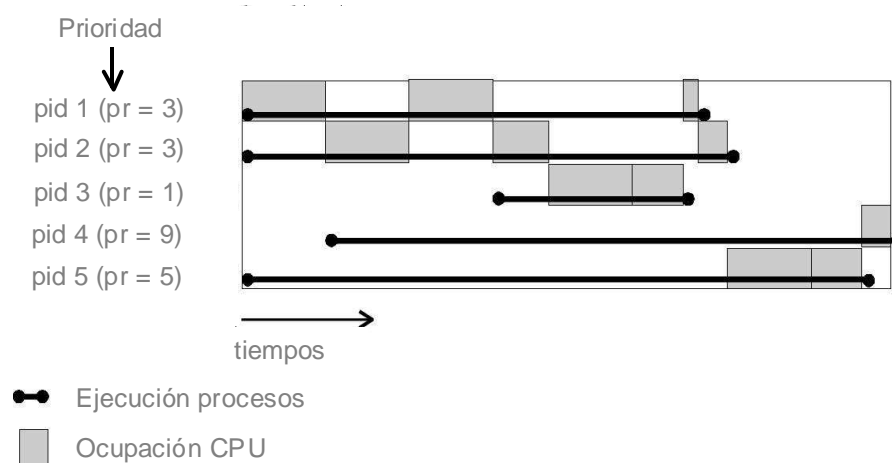


PAC1

Exercici 1 [3 punts]

Els sistemes operatius multiprocés poden incorporar en qualsevol moment, nous processos al conjunt de processos que ja s'estigui executant i competint per tant, per la CPU. Disposen, d'un mecanisme que els permet escollir quin procés podrà usar la CPU i durant quant de temps. El mecanisme té en compte el nivell de prioritat que s'assigna a cada procés en el moment que s'inicia l'execució. Suposem que estem treballant amb un sistema operatiu que treballa amb 10 nivells de prioritat per procés, de 0 a 9, sent 0 la més alta i 9 la més baixa.

El sistema operatiu assignarà la CPU de la manera següent: seleccionarà un procés en execució (a continuació veurem com) i l'executarà durant un interval de temps limitat. Si el procés acaba, es passarà a seleccionar un altre procés. Si no acaba, es desassignarà de la CPU, es tornarà a posar a la seqüència de processos que esperen i es seleccionarà un altre procés a ser servit (és a dir, ser executat).



Quan el sistema operatiu hagi de seleccionar un procés a servir, seleccionarà aquell que té la prioritats més alta (més propera a 0). Si n'hi ha més d'un, podrà seleccionar qualsevol d'ells (advertiu que això és una simplificació de la realitat, on normalment les polítiques per seleccionar el procés solen ser més complexes).

D'aquesta manera, veiem que un procés amb una certa prioritats no podrà ser servit a no ser que hagin finalitzat els processos que tenen més prioritats que ell. D'altra banda, també és possible eliminar un procés de la cua d'espera.

Es volen tenir les següent funcionalitats:

- Crear la seqüència buida.
- Executar un procés, és a dir, afegir-lo a la seqüència donada, a més de la informació del procés (per exemple, la direcció de memòria a partir de la qual resideix el codi a



executar), la seva prioritat i el temps de CPU que necessita. S'ha de retornar un identificador únic, anomenat PID - Process Identifier -, que identifica el procés en la seqüència.

- Enviar un procés a la CPU per a què es serveixi.
- Tornar un procés a espera quan ha acabat el temps màxim d'ocupació de la CPU.
- Eliminar un procés quan acaba la seva execució.
- Eliminar un procés identificat per PID de la seqüència.
- Permetre fer un llistat per pantalla dels processos que estan en espera ordenats de més a menys prioritat (a igual prioritat no importa l'ordre).

El TAD no decideix durant quant de temps es serveix un procés ni si un procés acaba la seva execució o es torna a la seqüència de processos. Això es decideix de manera externa al nostre TAD i per tant no ha de ser tractat.

Atès que el nombre de processos en espera no serà més gran que N, es demana:

Apartat (1.1) Defineix la signatura (operacions i sintaxi) del TAD i situa'l en la jerarquia de classes de la biblioteca de TADs de l'assignatura (d'on s'hauria d'estendre, relacions amb altres, etc.). Especialitza (substitueix) els paràmetres de les classes parametritzades de la jerarquia, sempre que et sigui possible, pels tipus que consideris més adients.

Raona la teva resposta.

Apartat 1.2) Fes l'especificació del TAD. Utilitza una especificació com la de l'apartat 1.2.3. Es valorarà especialment la concisió (absència d'elements redundants o innecessaris), precisió (definició correcta del resultat de les operacions), completesa (consideració de tots els casos possibles en què es pot executar cada operació) i manca d'ambigüitats (coneixement exacte de com es comporta cada operació en tots els casos possibles) de la solució. És important respondre aquest apartat usant una descripció condicional i no procedimental. L'experiència ens demostra que no sempre resulta fàcil distingir entre les dues descripcions, és per això que fem especial èmfasi insistint que poseu molta atenció a les vostres definicions. A títol d'exemple indicarem que la descripció condicional (la correcta a utilitzar en el contracte) d'omplir un got buit amb aigua seria:

@ pre el got es troba buit.

@ post el got és ple d'aigua.

En canvi una descripció procedimental (incorrecta per utilitzar en el contracte) tindria una forma semblant a:

@ pre el got hauria de trobar-se buit, si no es trobés buit s'hauria de buidar.

@ post s'acosta el got a l'aixeta i s'hi tira aigua fins que estigui ple.

Cal també tenir en compte que un contracte hauria de disposar d'invariant sempre que aquesta fos necessària per descriure el TAD.



PAC1 Estructura de la Informació curs 2010/2011 1r semestre de FUOC està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Exercici 2 [2,5 punts]

Es vol implementar un algorisme per avaluar un polinomi de grau n . Els coeficients del polinomi es guarden en un vector de $n + 1$ elements. Per exemple, un polinomi de grau 2 com $2x^2 + 3x - 5$ es guardaria en un vector de tres posicions $V = [2, 3, -5]$. A continuació es mostra una solució al problema:

```
(1) long evalua(int[] vector, int x, int grau) {
(2)     long s, pot;
(3)     int i, j;
(4)     s=0;
(5)     i=0;
(6)     while(i<=grau) {
(7)         j=0;
(8)         pot=1;
(9)         while(j< i) {
(10)             pot *= x;
(11)             j++;
(12)         }
(13)         s+=pot*vector[i];
(14)         i++;
(15)     }
(16)     return s;
(17)}
```

Es demana:

Apartat 2.1) Indicar detalladament la complexitat asimptòtica de la solució proposada.

Apartat 2.2) Proposar un algorisme que millori l'eficiència asimptòtica temporal de la solució proposada, indicant la eficiència (no cal fer el càlcul detallat).

Exercici 3 [1 punt]

Quina estructura de dades utilitzaries per representar una operació aritmètica amb parèntesis, per exemple $2 * (4 + 5 / (2 + 1) * 4) + 1$. Raoneu-ho i feu un esquema de com quedaria l'expressió anterior. Quin seria el recorregut més adequat per avaluar el resultat de l'operació?



PAC1 Estructura de la Informació curs 2010/2011 1r semestre de FUOC està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

Exercici 4 [1,5 punts]

Ordena les següents cotes de complexitat de menor a major, indicant també les que són iguals: $O(0.5 \cdot 2^n)$, $O(n+1)$, $O(2^{n/2})$, $O(\log_{10} n)$, $O((n+1)^2)$, $O(n \cdot \log n)$, $O(2 \log_6 n)$, $O(n^n)$, $O(3n^2)$, $O(n!)$, $O(n(n+\log n))$, $O(1)$, $O(1+1/n)$, $O(n^{n-2})$. Justifica breument les relacions de l'ordre obtingut.

Exercici 5 [2 punts]

Apartat 5.1) Ja coneixeu la representació de les cues mitjançant vectors i amb una estructura seqüencial, i també s'han estudiat una sèrie de millores a través de la utilització d'una gestió circular de les mateixes. Enumereu breument els avantatges i/o inconvenients que representaria una gestió circular en la implementació de les piles.

Apartat 5.2) En les llistes ordenades implementades mitjançant vectors podem millorar l'eficiència de la consulta mitjançant una recerca dicotòmica. En la implementació encadenada no es pot millorar l'eficiència d'aquesta manera, és necessari realitzar una cerca començant pel primer element. Comentar la millora en l'eficiència que obtindríem si per fer la cerca disposéssim d'un punter a l'element que ocupa la posició mitjana de la llista.

Apartat 5.3) Hem vist el funcionament de la implementació amb pilons dels arbres binaris. Creieu possible una implementació per piló d'arbres ternaris? Si ho creieu possible, de quina dimensió s'ha de crear el vector si volem emmagatzemar un arbre quasicomplet de X elements?. Proposeu un exemple.

Apartat 5.4) Ordenar el vector mitjançant l'algorisme de heap-sort utilitzant la versió que treballa amb un sol vector. Mostreu tots els estats intermedis del vector i la situació dels elements en cada pas:

4	1	7	2	6	3
---	---	---	---	---	---

