Sistemas operativos

Examen Diciembre 2020: Ejercicio 4

Resuelva el ejercicio planteado en uno solo de los siguientes enunciados.

Enunciado A)
Se quiere modelar una peluquería. Esta peluquería cuenta con 4 peluqueros, una sala de espera con capacidad para 20 clientes y una caja registradora con capacidad para 1 cliente. Si no hay clientes los peluqueros se duermen una siesta. Si un cliente entra a la peluquería y la capacidad de la sala de espera está colmada, se retira. En cambio, si los peluqueros están ocupados cortando y hay capacidad disponible en la sala de espera, entonces el cliente entra y espera. Si un peluquero está durmiendo el cliente lo despierta para que le corte el pelo. Los clientes deben atenderse en orden de llegada. Una vez finalizado el corte el cliente debe pagar en la caja registradora y su peluquero no puede cortarle al siguiente hasta que no pague.

Cuenta con las siguientes funciones:

- cortar_pelo(): invocada por el peluquero para cortar el pelo del cliente

- pagar_corte(): invocada por el cliente para pagar

Se pide: Implemente utilizando monitores los procesos peluquero y cliente.

Solución:

```
monitor Peluqueria {
    condition pelu_dormido;
    condition pelu_espera_pago[4];
    condition cli_espera;
    condition cli_cortando_pelo[4]

    int cant_clientes=0;
     bool pelu_listo=false;
     int mon_pelu_id;

     int cli_llega() {
      int pelu_id=-1;

     if (cant_clientes < 20) {
           cant_clientes++;
           pelu_dormido.signal();
           if (!pelu_listo)
                cli_espera.wait();
           cant_clientes--;
           pelu_listo=false;
           pelu_id=mon_pelu_id;
           cli_cortando_pelo[pelu_id].wait();
     }
```

```
      return pelu_id;
      }

      void cli_termina(int pelu_id) {
      pelu_espera_pago[pelu_id].signal();
      }

      void pelu_siguiente(int pelu_id) {
      if (cant_clientes > 0) {
           mon_pelu_id=pelu_id;
           cli_espera.signal();
      } else {
           pelu_dormido.wait();
           mon_pelu_id=pelu_id;
           pelu_listo=true;
           cli_espera.signal();
      }
      }

      pelu_espera_pago(int pelu_id) {
      cli_cortando_pelo[pelu_id].signal();
      pelu_espera_pago[pelu_id].wait();
      }
}

monitor Caja {
      pagar() {
      pagar_corte();
      }
}

void peluquero(int pelu_id) {
    while (true) {
       Peluqueria.pelu_siguiente(pelu_id)
       corta_pelo()
       Peluqueria.pelu_espera_pago(pelu_id)
      }
}

void cliente() {
      int pelu_id = Peluqueria.cli_llega()
      if (pelu_id <> -1) {
       Caja.pagar()
       Peluqueria.cli_termina(pelu_id)
      }
}

void main() {
    cobegin {
      peluquero(0); peluquero(1); peluquero(2); peluquero(3);
      cliente(); ... cliente();
    }
}
```

Enunciado B):
Se quiere modelar una peluquería. La peluquería cuenta con 2 peluqueros y una sala de espera con capacidad para 10 clientes. Si un cliente entra a la peluquería y la capacidad de la sala de espera está colmada, se retira. En cambio, si los peluqueros están ocupados cortando y hay capacidad disponible en la sala de espera, entonces el cliente entra y espera. Los clientes deben atenderse en orden de llegada. Los peluqueros siempre cortan el pelo juntos y hacen pasar de a dos clientes a la vez (nunca pasa un solo cliente). Les cortan el pelo (uno a cada uno) y luego los clientes se retiran al mismo tiempo. Luego de que ambos clientes se van y antes de ver si hay más clientes, los peluqueros chequean sus celulares y revisan sus redes sociales. Si un peluquero está listo pero el otro sigue chequeando su celular, entonces el que está listo duerme una siesta mientras espera. Si luego de que ambos terminan de revisar sus celulares resulta que no hay clientes en la sala de espera, entonces ambos peluqueros se duermen una siesta.

Cuenta con las siguientes funciones:

- cortar_pelo(): invocada por el peluquero para cortar el pelo del cliente.

- usar_celular(): invocada por el peluquero para revisar sus redes sociales.

Se pide: Implemente utilizando monitores los procesos peluquero y cliente.

Solución:

```
monitor Peluqueria {
    condition pelu_dormido;
    condition cli_espera;
    condition cli_cortando_pelo[3]

    int cant_espera=0;
    int cant_cortando=0;
    bool pelu_listo;
    int mon_pelu_id;

    void cli_espera_y_corte() {
        if (cantidad_clientes < 15) {
            cant_espera++
            if (cant_cortando == 2) {
                cli_espera.wait();
            } else {
                pelu_dormido.signal();
                if (!pelu_listo) {
                    cli_espera.wait();
                }
            }
            cant_espera--;
            pelu_listo=false;
            cli_cortando_pelo[mon_pelu_id].wait();
        }
```

```
    }

    void pelu_siguiente(int pelu_id) {
        if (cant_espera > 0 && cant_cortando < 2) {
            mon_pelu_id=pelu_id;
            cant_cortando++;
            cli_espera.signal();
        } else {
            pelu_dormido.wait()
            mon_pelu_id=pelu_id;
            pelu_listo=true;
            cant_cortando++;
            cli_espera.signal();
        }
    }

    void pelu_termine_corte(int pelu_id) {
        cli_cortando_pelo[pelu_id].signal();
        cant_cortando--;
        if (cant_espera > 0) {
            pelu_dormido.signal();
        }
    }
}

void peluquero(int pelu_id) {
     while (true) {
          Peluqueria.pelu_siguiente(pelu_id);
          cortar_pelo();
          Peluqueria.pelu_termine_corte(pelu_id)
    }
}

void cliente() {
    Peluqueria.cli_espera_y_corte();
}

void main() {
     cobegin {
          peluquero(0); peluquero(1); peluquero(2);
          cliente(); ... cliente();
     }
}
```

Enunciado C):
Se quiere modelar una peluquería. La peluquería cuenta con 3 peluqueros, un sector con capacidad para 2 clientes donde los peluqueros cortan el pelo, y una sala de espera con capacidad para 15 clientes. Si no hay clientes los peluqueros se duermen una siesta. Si un cliente entra a la peluquería y la capacidad de la sala de espera está colmada, se retira. En cambio, si los peluqueros están ocupados cortando y hay capacidad disponible en la sala de espera, entonces el cliente entra y espera. Si un peluquero está durmiendo y hay capacidad disponible para que el peluquero trabaje, el cliente lo despierta para que le corte el pelo. Los clientes deben atenderse en orden de llegada. Los peluqueros deben turnarse para trabajar.

Cuenta con las siguiente función:

- cortar_pelo(): invocada por el peluquero para cortar el pelo del cliente

Se pide: Implemente utilizando monitores los procesos peluquero y cliente.

Solución:

```
monitor Peluqueria {
    condition pelu_dormido;
    condition pelu_espera;
    condition cli_espera;
    condition cli_cortando_pelo;

    int cant_clientes=0;
    int cant_pelu_pronto=0;
    int cant_pelu_termine=0;

     void cli_espera_y_corte() {
     if (cant_clientes < 10) {
           cant_clientes++;
           if (cant_clientes <> 2) {
                 cli_espera.wait();
            } else {
                 if (cant_pelu_pronto == 2) {
                       cli_espera.signal();
                       pelu_dormido.signal();
                       pelu_dormido.signal();
                 } else {
                       cli_espera.wait();
                 }
            }
           cant_clientes--;
           cli_cortando_pelo.wait();
     }
     }

     void pelu_siguiente() {
      cant_pelu_pronto++;
      if (cant_pelu_pronto == 1) {
            pelu_dormido.wait();
     } else {
           if (cant_clientes > 2) {
```

```
                cli_espera.signal();
                cli_espera.signal();
                pelu_dormido.signal();
           } else {
                pelu_dormido.wait();
           }
       }
       cant_pelu_pronto--;
      }

      void pelu_terminar_corte() {
       cant_pelu_termine++;
       if (cant_pelu_termine == 1) {
            pelu_espera.wait();
       } else {
            cli_cortando_pelo.signal();
            cli_cortando_pelo.signal();
            pelu_espera.signal();
       }
       cant_pelu_termine--;
      }
}

void peluquero() {
      while (true) {
       Peluqueria.pelu_siguiente();
       corta_pelo();
       Peluqueria.pelu_terminar_corte();
       usar_celular();
      }
}

void cliente() {
      Peluqueria.cli_espera_y_corte();
}

void main() {
    cobegin {
      peluquero(); peluquero();
      cliente(); ... cliente();
    }
}
```