اثبات نیازمندی‌های اثبات سیستم (اثبات سازگاری سیستم)

در این بخش بر آنیم که آن بخش‌هایی را که نرم‌افزار AtelierB قادر به اثبات آن‌ها نبوده است را با استفاده از دستیار اثبات محاوره‌ایی این نرم‌افزار اثبات کنیم. دستیار اثبات از زبان مخصوصی استفاده می‌کند که توضیح این زبان از حوزه این پروژه خارج است و می‌توانید توضیح مفصل آن را در مجموعه اسناد [CLS012] مشاهده کنید. در ادامه اثبات تک تک نیازمندی‌های اثباتی را که ابزار AtelierB برای نشان دادن درستی سیستم استخراج کرده بود را مشاهده می‌کنیم.

اثبات‌های مربوط به بند INITIALISATION :

نیازمندی اثبات شماره ۱:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is established by
the initialisation - ref 3.3'"
=>
{running|->[root_server],ready|->{},blocked|->{}}: {running,ready,blocked} +->
iseq({root_server})
```

اثبات:

```
ff(0) & dd & ah([root_server]: iseq({root_server})) & ar(thEntire.1,Once) & pr & pr & dd
& pr
```

نیازمندی اثبات شماره ۲:

```
"`Check that the invariant (address_space_maplets: address_space -->
seq1(maplet_codomain)) is established by the initialisation - ref 3.3'"
=>
        {root_server_as|->mp}: {root_server_as} +-> seq(mc)-{{}}
```

اثبات:

```
ff(0) & dd & pr & ah(card(mc)>=1) & pr & ar(thEntire.2,Once) & pr & pr & dd & ah(ran(mp)
= mc) & dd & ar(thEntire.3,Once) & pr & pr & pr
```

نیازمندی اثبات شماره ۳:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is established by the initialisation - ref 3.3'"
=>
        UNION(qq).(qq: {running,ready,blocked} | ran({running|->[root_server],ready|-
>{},blocked|->{}}(qq))) = {root_server}
```

اثبات:

```
ff(0) & dd & ah(UNION(qq).(qq: {running,ready,blocked} | ran({running|-
>[root_server],ready|->{},blocked|->{}}(qq))) = ran({running|->[root_server],ready|-
>{},blocked|->{}}(ready))\/(ran({running|->[root_server],ready|->{},blocked|-
>{}}(running))\/ran({running|->[root_server],ready|->{},blocked|->{}}(blocked)))) &
ar(thEntire.5,Once) & dd & pr & ar(thEntire.4,Once) & pr & pr
```

نیازمندی اثبات شماره ٤:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
established by the initialisation - ref 3.3'"
=>
```

```
        INTER(qq).(qq: {running,ready,blocked} | ran({running|->[root_server],ready|-
>{},blocked|->{}}(qq))) = {}
```

اثبات:

```
ff(0) & dd & ah(INTER(qq).(qq: {running,ready,blocked} | ran({running|-
>[root_server],ready|->{},blocked|->{}}(qq))) = ran({running|->[root_server],ready|-
>{},blocked|->{}}(ready))/\(ran({running|->[root_server],ready|->{},blocked|-
>{}}(running))/\ran({running|->[root_server],ready|->{},blocked|->{}}(blocked)))) &
ar(thEntire.6,Once) & dd & pr
```

نیازمندی اثبات شماره ۵:

```
"`Check that the invariant (!as.(as: address_space =>
card(address_space_maplets(as))<=max_pg)) is established by the initialisation - ref
3.3'"
=>
        card({root_server_as|->mp}(as))<=max_pg
```

اثبات:

```
ff(0) & dd & ah(as = root_server_as) & pr & ar(thEntire.7,Once) & pr & pr & pr & pr
```

نیازمندی اثبات شماره ٦:

```
"`Check that the invariant (!ival.(ival: ran(maplet_codomain_indirect)-
{null_nat_as_tuple} => (%(pg,as).(pg: NATURAL1 & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: NATURAL1 & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival))))) is established by the initialisation -
ref 3.3'"
=>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))({}(ival))<={root_server_as|->root_server_as_sz}((%(pg,as).(pg: INTEGER & 0<=pg &
not(pg = 0) & as: ADDRESS_SPACE | as))({}(ival)))
```

اثبات:

```
ff(0) & pr & ah(ran(mc*{null_nat_as_tuple}) = {null_nat_as_tuple}) & pr & pr
```

نیازمندی اثبات شماره ۷:

```
"`Check that the invariant (!as_obj.(as_obj: address_space =>
size(address_space_maplets(as_obj)) = address_space_size(as_obj))) is established by the
initialisation - ref 3.3'"
=>
        size({root_server_as|->mp}(as_obj)) = {root_server_as|-
>root_server_as_sz}(as_obj)
```

اثبات:

```
ff(0) & dd & ah(as_obj = root_server_as) & pr & ah(card(np) = card(mc)) &
ar(thEntire.8,Once) & pr & pr & dd & ah(size(mp) = card(mc)) & ar(thEntire.9,Once) & pr
& pr & dd & pr & pr
```

قواعد مورد استفاده در اثبات‌های مربوط به بند INITIALISATION

```
thEntire.1:
        a : seq (b) & size (a) = 1 => a : iseq (b);
thEntire.2:
        card (a) = root_server_as_sz & root_server_as_sz >= 1 => a /= {};
thEntire.3:
        a : seq (mc) & ran (a) = mc & card (mc) >= 1 => size (a) >= 1;
thEntire.4:
        a : seq (b) & size (a) = 1 => ran (a) = b;
thEntire.5:
        UNION(qq).(qq: {running,ready,blocked} | ran(f(qq))) = ran (f(ready)) \/ (ran
(f(running)) \/ ran (f(blocked)));
thEntire.6:
        INTER(qq).(qq: {running,ready,blocked} | ran(f(qq))) = ran (f(ready)) /\ (ran
(f(running)) /\ ran (f(blocked)));
```

```
thEntire.7:
        a : seq (mc) & ran (a) = mc & card (mc) <= max_pg => size (a) <= max_pg;
thEntire.8:
        card (a) = root_server_as_sz & card (b) = root_server_as_sz => card (a) = card
(b);
thEntire.9:
        a : seq (mc) & ran (a) = mc => size (a) = card (mc);
<<USELESS>> thEntire.10:
<<USELESS>>     card (a) >= 1 => a /= {};
```

اثبات‌های مربوط به عملگرهای sendMessage و receiveMessage :

نیازمندی اثبات شماره ۱:

```
"`Local hypotheses'" &
        current = first(queue_processes(running)) &
        "`Check that the invariant (queue_processes: queue --> iseq(process)) is
preserved by the operation - ref 3.4'"
=>
        queue_processes<+{running|->{},blocked|->(queue_processes(blocked)<-current)}:
queue +-> iseq(process)
```

اثبات:

```
ff(0) & pr & ah(current: ran(queue_processes(running))) & pr & dd &
ah(ran(queue_processes(running))/\ran(queue_processes(blocked)) = {}) &
ar(thEntire.11,Once) & pr & dd & cts & pr & pr
```

نیازمندی اثبات شماره ۲:

```
"`Local hypotheses'" &
        current = first(queue_processes(running)) &
        "`Check that the invariant (queue_processes: queue --> iseq(process)) is
preserved by the operation - ref 3.4'"
=>
        dom(queue_processes<+{running|->{},blocked|->(queue_processes(blocked)<-
current)}) = queue
```

اثبات:

```
ff(0) & dd & pr & pr
```

نیازمندی اثبات شماره ۳:

```
"`Check that the invariant
(size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the
operation - ref 3.4'"
=>
        1<=size((queue_processes<+{running|->{},blocked|->(queue_processes(blocked)<-
current)})(running))+size((queue_processes<+{running|->{},blocked|-
>(queue_processes(blocked)<-current)})(ready))
```

اثبات:

```
ff(0) & dd & ah(size((queue_processes<+{running|->{},blocked|-
>(queue_processes(blocked)<-current)})(running)) = 0) & pr &
ah((queue_processes<+{running|->{},blocked|->(queue_processes(blocked)<-
first(queue_processes(running)))})(running) = {running|->{},blocked|-
>(queue_processes(blocked)<-first(queue_processes(running)))}(running)) & ah(running:
dom({running|->{},blocked|->(queue_processes(blocked)<-
first(queue_processes(running)))})) & pr & dd & pr & dd & pr & dd & pr & pr &
ah(ready/:dom({running|->{},blocked|->(queue_processes(blocked)<-
first(queue_processes(running)))})) & pr & dd & pr
```

نیازمندی اثبات شماره ٤:

```
"`Check that the invariant (size(queue_processes(running))<=1) is preserved by the
operation - ref 3.4'"
=>
```

```
        size((queue_processes<+{running|->{},blocked|->(queue_processes(blocked)<-
current)})(running))<=1
```

اثبات:

```
ff(0) & dd & ah(running: dom({running|->{},blocked|->(queue_processes(blocked)<-
current)})) & pr & dd & pr
```

نیازمندی اثبات شماره ۵:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{running|->{},blocked|-
>(queue_processes(blocked)<-current)})(qq))) = process
```

اثبات:

```
ff(0) & dd & ah(UNION(qq).(qq: queue | ran((queue_processes<+{running|->{},blocked|-
>(queue_processes(blocked)<-current)})(qq))) =
ran({})\/ran(queue_processes(ready))\/ran(queue_processes(blocked)<-current)) & pr & pr
& ah({first(queue_processes(running))} = ran(queue_processes(running))) &
ar(thEntire.12,Once) & ar(thEntire.13,Once) & pr & pr & dd & pr & ar(thEntire.14,Once) &
dd & pr & ah(UNION(qq).(qq: dom(queue_processes) | ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & pr & ar(thEntire.5,Once) & dd & pr & pr & ah({first(queue_processes(running))} =
ran(queue_processes(running))) & ar(thEntire.12,Once) & ar(thEntire.13,Once) & pr & pr &
dd & pr
```

نیازمندی اثبات شماره ٦:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{running|->{},blocked|-
>(queue_processes(blocked)<-current)})(qq))) = {}
```

اثبات:

```
ff(0) & dd & pr & pr & ah(INTER(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{running|->{},blocked|->(queue_processes(blocked)<-
first(queue_processes(running)))})(qq))) = ran((queue_processes<+{running|->{},blocked|-
>(queue_processes(blocked)<-
first(queue_processes(running)))})(ready))/\(ran((queue_processes<+{running|-
>{},blocked|->(queue_processes(blocked)<-
first(queue_processes(running)))})(running))/\ran((queue_processes<+{running|-
>{},blocked|->(queue_processes(blocked)<-first(queue_processes(running)))})(blocked))))
& ar(thEntire.6,Once) & dd & pr & ah(ran((queue_processes<+{running|->{},blocked|-
>(queue_processes(blocked)<-first(queue_processes(running)))})(running)) = {}) &
ah(running: dom({running|->{},blocked|->(queue_processes(blocked)<-
first(queue_processes(running)))})) & pr & dd & pr & dd & pr
```

نیازمندی اثبات شماره ۷:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        queue_processes<+{ready|->(queue_processes(ready)<-pp),blocked|->blocked_q}:
queue +-> iseq(process)
```

اثبات:

```
ff(0) & dd & pr & ah(INTER(qq).(qq: queue | ran(queue_processes(qq))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & pr & pr & ar(thEntire.6,Once) & dd & ah(INTER(qq).(qq: queue |
ran(queue_processes(qq))) = {} & INTER(qq).(qq: queue | ran(queue_processes(qq))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
))) =>
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
))) = {}) & dd & ar(thEntire.15,Once) & pr & pr & pr &
ah(ran(queue_processes(running))/={}) & ar(thEntire.16,Once) & ar(thEntire.17,Once) & pr
```

```
& dd & ah(ran(queue_processes(ready))/\ran(queue_processes(blocked)) = {}) &
ah(ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(bloc
ked))) = ran(queue_processes(ready))/\ran(queue_processes(blocked))) &
ar(thEntire.18,Once) & pr & pr & dd & pr & ar(thEntire.19,Once) & pr & pr & pr & dd & pr
& pr & ah(ran(queue_processes(blocked))-{pp} <: UNION(qq).(qq: dom(queue_processes) |
ran(queue_processes(qq)))) & pr & pr & dd & pr & dd &
ah(seq(ran(queue_processes(blocked))-{pp}) <: seq(UNION(qq).(qq: dom(queue_processes) |
ran(queue_processes(qq))))) & ar(thEntire.20,Once) & pr & dd & ar(thEntire.21,Once) & pr
& pr & pr & pr & pr & pr & pr
```

نیازمندی اثبات شماره ۸:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        dom(queue_processes<+{ready|->(queue_processes(ready)<-pp),blocked|->blocked_q})
= queue
```

اثبات:

```
ff(0) & dd & pr & pr
```

نیازمندی اثبات شماره ۹:

```
"`Check that the invariant
(size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the
operation - ref 3.4'"
=>
        1<=size((queue_processes<+{ready|->(queue_processes(ready)<-pp),blocked|-
>blocked_q})(running))+size((queue_processes<+{ready|->(queue_processes(ready)<-
pp),blocked|->blocked_q})(ready))
```

اثبات:

```
ff(0) & dd & ah((queue_processes<+{ready|->(queue_processes(ready)<-pp),blocked|-
>blocked_q})(running) = queue_processes(running)) & ah(running/:dom({ready|-
>(queue_processes(ready)<-pp),blocked|->blocked_q})) & pr & dd & pr & dd &
ah(size((queue_processes<+{ready|->(queue_processes(ready)<-pp),blocked|-
>blocked_q})(running)) = size(queue_processes(running))) & pr & pr & dd &
ah(size(queue_processes(running))>0) & pr & dd & ah(size(queue_processes(running))<=1) &
dd & ah(size(queue_processes(running)) = 1) & ar(thEntire.13,Once) & pr & pr & dd & pr &
pr
```

نیازمندی اثبات شماره ۱۰:

```
"`Check that the invariant (size(queue_processes(running))<=1) is preserved by the
operation - ref 3.4'"
=>
        size((queue_processes<+{ready|->(queue_processes(ready)<-pp),blocked|-
>blocked_q})(running))<=1
```

اثبات:

```
ff(0) & dd & ah(running/:dom({ready|->(queue_processes(ready)<-pp),blocked|-
>blocked_q})) & pr & dd & pr
```

نیازمندی اثبات شماره ۱۱:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{ready|->(queue_processes(ready)<-
pp),blocked|->blocked_q})(qq))) = process
```

اثبات:

```
ff(0) & dd & pr & pr & ah(UNION(qq).(qq: {running,ready,blocked} |
ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
```

```
)))) & ar(thEntire.5,Once) & dd & pr & ah(UNION(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{ready|->(queue_processes(ready)<-process_pid~(pid)),blocked|-
>blocked_q})(qq))) = ran((queue_processes<+{ready|->(queue_processes(ready)<-
process_pid~(pid)),blocked|->blocked_q})(ready))\/(ran((queue_processes<+{ready|-
>(queue_processes(ready)<-process_pid~(pid)),blocked|-
>blocked_q})(running))\/ran((queue_processes<+{ready|->(queue_processes(ready)<-
process_pid~(pid)),blocked|->blocked_q})(blocked)))) & ar(thEntire.5,Once) & dd & pr &
ah(ran((queue_processes<+{ready|->(queue_processes(ready)<-process_pid~(pid)),blocked|-
>blocked_q})(running)) = ran(queue_processes(running))) & ah(running/:dom({ready|-
>(queue_processes(ready)<-process_pid~(pid)),blocked|->blocked_q})) & pr & dd & pr & dd
& pr & ah(ran((queue_processes<+{ready|->(queue_processes(ready)<-
process_pid~(pid)),blocked|->blocked_q})(ready))\/ran((queue_processes<+{ready|-
>(queue_processes(ready)<-process_pid~(pid)),blocked|->blocked_q})(blocked)) =
ran(queue_processes(ready))\/ran(queue_processes(blocked))) & ah(ready: dom({ready|-
>(queue_processes(ready)<-process_pid~(pid)),blocked|->blocked_q})) & pr & dd &
ah(blocked: dom({ready|->(queue_processes(ready)<-process_pid~(pid)),blocked|-
>blocked_q})) & pr & dd & pr & dd & ar(thEntire.22,Once) & pr
```

نیازمندی اثبات شماره ۱۲:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{ready|->(queue_processes(ready)<-
pp),blocked|->blocked_q})(qq))) = {}
```

اثبات:

```
ff(0) & dd & pr & pr & ah(INTER(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{ready|->(queue_processes(ready)<-process_pid~(pid)),blocked|-
>blocked_q})(qq))) = ran((queue_processes<+{ready|->(queue_processes(ready)<-
process_pid~(pid)),blocked|->blocked_q})(ready))/\(ran((queue_processes<+{ready|-
>(queue_processes(ready)<-process_pid~(pid)),blocked|-
>blocked_q})(running))/\ran((queue_processes<+{ready|->(queue_processes(ready)<-
process_pid~(pid)),blocked|->blocked_q})(blocked)))) & ar(thEntire.6,Once) & dd &
ah(ran((queue_processes<+{ready|->(queue_processes(ready)<-process_pid~(pid)),blocked|-
>blocked_q})(ready))/\(ran((queue_processes<+{ready|->(queue_processes(ready)<-
process_pid~(pid)),blocked|->blocked_q})(running))/\ran((queue_processes<+{ready|-
>(queue_processes(ready)<-process_pid~(pid)),blocked|->blocked_q})(blocked))) = {}) &
ah(ready: dom({ready|->(queue_processes(ready)<-process_pid~(pid)),blocked|-
>blocked_q})) & pr & dd & ah(running/:dom({ready|->(queue_processes(ready)<-
process_pid~(pid)),blocked|->blocked_q})) & pr & dd & ah(blocked: dom({ready|-
>(queue_processes(ready)<-process_pid~(pid)),blocked|->blocked_q})) & pr & dd &
ah(ran((queue_processes<+{ready|->(queue_processes(ready)<-process_pid~(pid)),blocked|-
>blocked_q})(ready))/\(ran((queue_processes<+{ready|->(queue_processes(ready)<-
process_pid~(pid)),blocked|->blocked_q})(running))/\ran((queue_processes<+{ready|-
>(queue_processes(ready)<-process_pid~(pid)),blocked|->blocked_q})(blocked))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & pr & ar(thEntire.23,Once) & dd & ar(thEntire.24,Once) & pr & dd &
ah(INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) & dd & ar(thEntire.15,Once) &
pr & pr & pr & ar(thEntire.26,Once) & pr & ar(thEntire.15,Once) & pr & ah(queue =
{running,ready,blocked}) & ar(thEntire.6,Once) & pr & dd & pr
```

قواعد مورد استفاده در اثبات‌های مربوط به عملگرهای sendMessage و receiveMessage

```
thEntire.11:
        INTER(qq).(qq: queue | ran(queue_processes(qq))) = {} => ran
(queue_processes(running)) /\ ran (queue_processes(blocked)) = {};
thEntire.12:
        size (a) = 1 => {first (a)} = ran (a);
thEntire.13:
        size (a) > 0 & size (a) <= 1 => size (a) = 1;
thEntire.14:
        UNION(qq).(qq: {running,ready,blocked} | ran((queue_processes<+{running|-
>{},blocked|->(queue_processes(blocked)<-first(queue_processes(running)))})
            (qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)));
thEntire.15:
        INTER(qq).(qq: queue | ran(queue_processes(qq))) = {} &
        INTER(qq).(qq: queue | ran(queue_processes(qq))) = b /\ (c /\ d)
        =>
        b /\ (c /\ d) = {};
```

```
thEntire.16:
        a /= {} => ran (a) /= {};
thEntire.17:
        size (a) > 0 => a /= {};
thEntire.18:
        a /\ (b /\ c) = {} &
        b /= {}
        =>
        a /\ (b /\ c) = a /\ c;
thEntire.19:
        ran(queue_processes(running)) /= {} &
        a/\(ran(queue_processes(running))/\c) = a/\c &
        a/\(ran(queue_processes(running))/\c) = {}
        =>
        a/\c = {};
thEntire.20:
        a <: b => seq (a) <: seq (b);
thEntire.21:
        a : seq(ran(queue_processes(blocked))-{pp}) &
        seq(ran(queue_processes(blocked))-{pp}) <: b
        =>
        a : b;
thEntire.22:
        a \/ c = d \/ f
        =>
        a \/ (b \/ c) = d \/ (b \/ f);
thEntire.23:
        ({a} \/ b /\ (c /\ d)) - {a} = b /\ (c /\ d);
thEntire.24:
        a /: ({a} \/ b /\ (c /\ d))
        =>
        {a} \/ b /\ (c /\ d) = b /\ (c /\ d);
thEntire.26:
        a /\ (b /\ ran(queue_processes(blocked))) = {} &
        d = ran(queue_processes(blocked)) - {pp}
        =>
        a /\ (b /\ d) = {};
```

اثبات‌های مربوط به عملگر schedule :

نیازمندی اثبات شماره ۱:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        queue_processes<+{running|->[first(queue_processes(ready))],ready|-
>tail(queue_processes(ready))}: queue +-> iseq(process)
```

اثبات:

```
ff(0) & dd & pr & ah(first(queue_processes(ready)): process) & dd & ah(UNION(qq).(qq:
dom(queue_processes) | ran(queue_processes(qq))) = process) & dd & ah(UNION(qq).(qq:
{running,ready,blocked} | ran(queue_processes(qq))) = process) & pr & dd & pr &
ar(thEntire.25,Goal) & pr & ah([first(queue_processes(ready))]: iseq(process)) &
ar(thEntire.1,Once) & pr & pr & dd & pr & pr & pr
```

نیازمندی اثبات شماره ۲:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        dom(queue_processes<+{running|->[first(queue_processes(ready))],ready|-
>tail(queue_processes(ready))}) = queue
```

اثبات:

```
ff(0) & dd & pr & pr
```

نیازمندی اثبات شماره ۳:

```
"`Check that the invariant
```

```
(size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the
operation - ref 3.4'"
=>
        1<=size((queue_processes<+{running|->[first(queue_processes(ready))],ready|-
>tail(queue_processes(ready))})(running))+size((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|->tail(queue_processes(ready))})(ready))
```

اثبات:

```
ff(0) & dd & ah(running: dom({running|->[first(queue_processes(ready))],ready|-
>tail(queue_processes(ready))})) & pr & dd & ah((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|->tail(queue_processes(ready))})(running) =
[first(queue_processes(ready))]) & pr & dd & ah(size([first(queue_processes(ready))]) =
1) & pr & dd & pr
```

نیازمندی اثبات شماره ٤:

```
"`Check that the invariant (size(queue_processes(running))<=1) is preserved by the
operation - ref 3.4'"
=>
        size((queue_processes<+{running|->[first(queue_processes(ready))],ready|-
>tail(queue_processes(ready))})(running))<=1
```

اثبات:

```
ff(0) & dd & ah(running: dom({running|->[first(queue_processes(ready))],ready|-
>tail(queue_processes(ready))})) & pr & dd & ah((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|->tail(queue_processes(ready))})(running) =
[first(queue_processes(ready))]) & pr & dd & ah(size([first(queue_processes(ready))]) =
1) & pr & dd & pr
```

نیازمندی اثبات شماره ٥:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|->tail(queue_processes(ready))})(qq))) = process
```

اثبات:

```
ff(0) & dd & pr & pr & ah(UNION(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{running|->[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(qq))) = ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(ready))\/(ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(running))\/ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|->(queue_processes(ready)\|/1)})(blocked)))) &
ar(thEntire.5,Once) & dd & ah(ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(ready))\/(ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(running))\/ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|->(queue_processes(ready)\|/1)})(blocked))) =
ran(queue_processes(ready)\|/1)\/(ran([first(queue_processes(ready))])\/ran(queue_proces
ses(blocked)))) & ah(ready: dom({running|->[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})) & pr & dd & pr & ah(running: dom({running|-
>[first(queue_processes(ready))],ready|->(queue_processes(ready)\|/1)})) & pr & dd &
ah(blocked/:dom({running|->[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})) & pr & dd & pr & dd &
ah(ran(queue_processes(ready)\|/1)\/(ran([first(queue_processes(ready))])\/ran(queue_pro
cesses(blocked))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & ar(thEntire.27,Once) & pr & dd & pr & ah(UNION(qq).(qq: {running,ready,blocked} |
ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & ar(thEntire.5,Once) & dd & pr
```

نیازمندی اثبات شماره ٦:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|->tail(queue_processes(ready))})(qq))) = {}
```

اثبات:

```
ff(0) & dd & pr & pr & pr & pr & ah(INTER(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{running|->[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(qq))) = ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(ready))/\(ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(running))/\ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|->(queue_processes(ready)\|/1)})(blocked)))) &
ar(thEntire.6,Once) & dd & ah(ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(ready))/\(ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})(running))/\ran((queue_processes<+{running|-
>[first(queue_processes(ready))],ready|->(queue_processes(ready)\|/1)})(blocked))) = {})
& ah(running: dom({running|->[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})) & pr & dd & ah(ready: dom({running|-
>[first(queue_processes(ready))],ready|->(queue_processes(ready)\|/1)})) & pr & dd &
ah(blocked/:dom({running|->[first(queue_processes(ready))],ready|-
>(queue_processes(ready)\|/1)})) & pr & dd & pr &
ah(ran([first(queue_processes(ready))]) = {first(queue_processes(ready))}) &
ar(thEntire.28,Once) & dd & ah(ran(queue_processes(ready)\|/1) =
ran(queue_processes(ready))-{first(queue_processes(ready))}) & pr & ar(thEntire.29,Once)
& dd & ah(ran([first(queue_processes(ready))]) = {first(queue_processes(ready))}) & dd &
ah(ran(queue_processes(ready)\|/1)/\(ran([first(queue_processes(ready))])/\ran(queue_pro
cesses(blocked))) = ran(queue_processes(ready))-
{first(queue_processes(ready))}/\({first(queue_processes(ready))}/\ran(queue_processes(b
locked)))) & pr & dd & ah(ran(queue_processes(ready))-
{first(queue_processes(ready))}/\({first(queue_processes(ready))}/\ran(queue_processes(b
locked))) = {}) & ar(thEntire.30,Once) & dd & pr & dd & pr
```

نیازمندی اثبات شماره ۷:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        queue_processes<+{ready|->tail(queue_processes(ready)<-
first(queue_processes(running))),running|->[first(queue_processes(ready))]}: queue +->
iseq(process)
```

اثبات:

```
ff(0) & dd & pr & pr & ar(thEntire.25,Goal) & pr & ah(ran(queue_processes(ready)) <:
process) & dd & ah(first(queue_processes(running)): process) & dd &
ah(ran(queue_processes(ready)<-first(queue_processes(running))) <: process) & pr & dd &
pr & ah(queue_processes(ready)<-first(queue_processes(running))\|/1: iseq(process)) &
ah(queue_processes(ready): iseq(process)) & pr & dd & ah(queue_processes(ready)<-
first(queue_processes(running)): iseq(process)) & pr & ct &
ah(ran(queue_processes(ready))/\ran(queue_processes(running))/={}) &
ar(thEntire.31,Once) & pr & dd & ah(INTER(qq).(qq: queue | ran(queue_processes(qq))) =
{}) & dd & ah(INTER(qq).(qq: queue | ran(queue_processes(qq))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & pr & pr & ar(thEntire.6,Once) & dd & ar(thEntire.32,Once) & pr & pr & pr & dd &
ar(thEntire.34,Once) & pr & dd & ar(thEntire.35,Once) & pr &
ah([first(queue_processes(ready))]: iseq(process)) & ar(thEntire.36,Once) & pr & dd & pr
& pr
```

نیازمندی اثبات شماره ۸:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        dom(queue_processes<+{ready|->tail(queue_processes(ready)<-
first(queue_processes(running))),running|->[first(queue_processes(ready))]}) = queue
```

اثبات:

```
ff(0) & dd & pr & pr
```

نیازمندی اثبات شماره ۹:

```
"`Check that the invariant
(size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the
operation - ref 3.4'"
=>
        1<=size((queue_processes<+{ready|->tail(queue_processes(ready)<-
first(queue_processes(running))),running|-
>[first(queue_processes(ready))]})(running))+size((queue_processes<+{ready|-
>tail(queue_processes(ready)<-first(queue_processes(running))),running|-
>[first(queue_processes(ready))]})(ready))
```

اثبات:

```
ff(0) & dd & ah(running: dom({ready|->tail(queue_processes(ready)<-
first(queue_processes(running))),running|->[first(queue_processes(ready))]})) & pr & dd
& ah((queue_processes<+{ready|->tail(queue_processes(ready)<-
first(queue_processes(running))),running|->[first(queue_processes(ready))]})(running) =
[first(queue_processes(ready))]) & pr & dd & ah(size([first(queue_processes(ready))]) =
1) & pr & dd & pr
```

نیازمندی اثبات شماره ۱۰:

```
"`Check that the invariant (size(queue_processes(running))<=1) is preserved by the
operation - ref 3.4'"
=>
        size((queue_processes<+{ready|->tail(queue_processes(ready)<-
first(queue_processes(running))),running|-
>[first(queue_processes(ready))]})(running))<=1
```

اثبات:

```
ff(0) & dd & ah(running: dom({ready|->tail(queue_processes(ready)<-
first(queue_processes(running))),running|->[first(queue_processes(ready))]})) & pr & dd
& ah((queue_processes<+{ready|->tail(queue_processes(ready)<-
first(queue_processes(running))),running|->[first(queue_processes(ready))]})(running) =
[first(queue_processes(ready))]) & pr & dd & ah(size([first(queue_processes(ready))]) =
1) & pr & dd & pr
```

نیازمندی اثبات شماره ۱۱:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{ready|-
>tail(queue_processes(ready)<-first(queue_processes(running))),running|-
>[first(queue_processes(ready))]})(qq))) = process
```

اثبات:

```
ff(0) & dd & pr & pr & ah(UNION(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{ready|->(queue_processes(ready)<-
first(queue_processes(running))\|/1),running|->[first(queue_processes(ready))]})(qq))) =
ran((queue_processes<+{ready|->(queue_processes(ready)<-
first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(ready))\/(ran((queue_processes<+{ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(running))\/ran((queue_processes<+{ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(blocked)))) & ar(thEntire.5,Once) & dd &
ah(ran((queue_processes<+{ready|->(queue_processes(ready)<-
first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(ready))\/(ran((queue_processes<+{ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(running))\/ran((queue_processes<+{ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(blocked))) = UNION(qq).(qq: {running,ready,blocked} |
```

```
ran(queue_processes(qq))) & ah(ready: dom({ready|->(queue_processes(ready)<-
first(queue_processes(running))\|/1),running|->[first(queue_processes(ready))]})) & pr &
dd & ah(running: dom({ready|->(queue_processes(ready)<-
first(queue_processes(running))\|/1),running|->[first(queue_processes(ready))]})) & pr &
dd & ah(blocked/:dom({ready|->(queue_processes(ready)<-
first(queue_processes(running))\|/1),running|->[first(queue_processes(ready))]})) & pr &
dd & pr & ah(ran(queue_processes(ready)<-first(queue_processes(running))\|/1) =
(ran(queue_processes(ready))\/ran(queue_processes(running)))-
{first(queue_processes(ready))}) & ar(thEntire.37,Once) & pr & pr & dd &
ah(ran([first(queue_processes(ready))]) = {first(queue_processes(ready))}) &
ar(thEntire.28,Once) & dd & pr &
ah((ran(queue_processes(ready))\/ran(queue_processes(running)))-
{first(queue_processes(ready))}\/({first(queue_processes(ready))}\/ran(queue_processes(b
locked))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & ar(thEntire.38,Once) & dd & pr & ah(UNION(qq).(qq: dom(queue_processes) |
ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & pr & pr & ar(thEntire.5,Once) & dd & pr & dd & pr
```

نیازمندی اثبات شماره ۱۲:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{ready|-
>tail(queue_processes(ready)<-first(queue_processes(running))),running|-
>[first(queue_processes(ready))]})(qq))) = {}
```

اثبات:

```
ff(0) & dd & pr & pr & ah(INTER(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{ready|->(queue_processes(ready)<-
first(queue_processes(running))\|/1),running|->[first(queue_processes(ready))]})(qq))) =
ran((queue_processes<+{ready|->(queue_processes(ready)<-
first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(ready))/\(ran((queue_processes<+{ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(running))/\ran((queue_processes<+{ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(blocked)))) & ar(thEntire.6,Once) & dd &
ah(ran((queue_processes<+{ready|->(queue_processes(ready)<-
first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(ready))/\(ran((queue_processes<+{ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(running))/\ran((queue_processes<+{ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})(blocked))) = {}) & ah(ready: dom({ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})) & pr & dd & ah(running: dom({ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})) & pr & dd & ah(blocked/:dom({ready|-
>(queue_processes(ready)<-first(queue_processes(running))\|/1),running|-
>[first(queue_processes(ready))]})) & pr & dd & pr & ar(thEntire.40,Once) &
ar(thEntire.37,Once) & pr & pr & ar(thEntire.28,Once) & dd & pr
```

قواعد مورد استفاده در اثبات‌های مربوط به عملگر schedule

```
thEntire.25:
        UNION(qq).(qq: {running,ready,blocked} | ran(queue_processes(qq))) = process
        =>
        UNION(qq).(qq: {running,ready,blocked} | ran(queue_processes(qq))) == process;
thEntire.27:
        b = {}
        =>
        ran (a\|/1) \/ (ran ([first (a)]) \/ ran (c)) = ran (a) \/ (ran (b) \/ ran (c))
thEntire.28:
        ran ([a]) = {a};
thEntire.29:
        ran(a\|/1) = ran (a) - {first (a)};
thEntire.30:
        ran (a) - {first (a)} /\ ({first (a)} /\ ran (b)) = {}
thEntire.31:
```

```
        first(b) : ran (a)
        =>
        ran (a) /\ ran (b) /= {};
thEntire.32:
        INTER(qq).(qq: queue | ran(queue_processes(qq))) = {} &
        INTER(qq).(qq: queue | ran(queue_processes(qq))) = ran (queue_processes(ready))
/\
        (ran (queue_processes(running)) /\ ran (queue_processes(blocked))) &
        ran (queue_processes(ready)) /\ ran (queue_processes(running)) /= {}
        =>
        bfalse;
thEntire.33:
        a : iseq (process) &
        ran (a) /\ ran (b) = {}
        =>
        a <- first (b) : iseq (process);
thEntire.34:
        a : iseq (process)
        =>
        a\|/1 : iseq (process);
thEntire.35:
        a : iseq (process)
        =>
        a~ : process +-> NATURAL;
thEntire.36:
        a : process
        =>
        [a] : iseq (process);
thEntire.37:
        b : seq (process) &
        size (b) <= 1
        =>
        ran (a <- first (b) \|/ 1) = (ran (a) \/ ran (b)) - {first (a)};
thEntire.38:
        (a \/ b) - {c} \/ ({c} \/ d) = a \/ (b \/ d);
thEntire.39:
        c : a
        =>
        (a \/ b) - {c} /\ ({c} /\ d) = {};
thEntire.40:
        ran (a <- first (b) \|/ 1) = (ran (a) \/ ran (b)) - {first (a)} &
        ran ([first(a)]) = {first(a)}
        =>
        ran (a <- first (b) \|/ 1) /\ (ran ([first(a)]) /\ c) = {}
```

اثبات‌های مربوط به عملگر **reclaimPage**:

نیازمندی اثبات شماره ۱:

```
"`Check that the invariant (maplet_codomain_indirect: maplet_codomain -->
nat_as_tuple\/{null_nat_as_tuple}) is preserved by the operation - ref 3.4'"
=>
        dom(maplet_codomain_indirect<+ml_to_be_null*{null_nat_as_tuple}) =
maplet_codomain
```

اثبات:

```
ff(0) & dd & pr & pr & pr & pr & ah(dom(maplet_codomain_indirect) =
ran(maplet_codomain_indirect~)) & pr & dd & ah(ran(maplet_codomain_indirect~) <:
dom(maplet_codomain_indirect)) & pr & dd & pr
```

نیازمندی اثبات شماره ۲:

```
"`Check that the invariant (!ival.(ival: ran(maplet_codomain_indirect)-
{null_nat_as_tuple} => (%(pg,as).(pg: NATURAL1 & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: NATURAL1 & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival))))) is preserved by the operation - ref
3.4'"
=>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))((ind_bl<<|nat_as_tuple_val)(ival))<=address_space_size((%(pg,as).(pg: INTEGER &
0<=pg & not(pg = 0) & as: ADDRESS_SPACE | as))((ind_bl<<|nat_as_tuple_val)(ival)))
```

اثبات:

```
ff(0) & dd & ah(!ival.(ival: ran(maplet_codomain_indirect)-{null_nat_as_tuple} =>
(%(pg,as).(pg: NATURAL1 & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: NATURAL1 & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival)))) & pr & dd & dc(ival/:ind_bl) & dd & pr &
ar(thEntire.46,Once) & pr & ar(thEntire.47,Once) & pr & pr & pr & dd &
ah(ival/:dom(ind_bl<<|nat_as_tuple_val)) & pr & dd & pr & ar(thEntire.46,Once) & pr &
ar(thEntire.47,Once) & pr & pr & pr
```

نیازمندی اثبات شماره ۳:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_indirect(ml)/=null_nat_as_tuple => maplet_codomain_real(ml) =
null_nat_ptr))) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_real(ml) = null_nat_ptr
```

اثبات:

```
ff(0) & dd & ah(ml/:ml_to_be_null) & ct & ah(ml: dom(ml_to_be_null*{null_nat_as_tuple}))
& pr & dd & ar(thEntire.50,Once) & pr & pr & dd &
ah(maplet_codomain_indirect(ml)/=null_nat_as_tuple) & ar(thEntire.48,Once) & pr & pr &
dd & ar(thEntire.49,Once) & pr & pr
```

قواعد مورد استفاده در اثبات‌های مربوط به عملگر reclaimPage

```
thEntire.41:
        ran (a) <: NATURAL1*address_space
        =>
        dom (ran (a)) <: NATURAL1;
thEntire.42:
        ran (a) <: NATURAL1*address_space &
        b : NATURAL1*address_space
        =>
        dom (ran (a) - {b}) <: NATURAL;
thEntire.43:
        a <: NATURAL1*address_space
        =>
        dom (a) <: NATURAL1;
thEntire.44:
        0 : dom(ran(ind_bl<<|nat_as_tuple_val)) &
        dom(ran(ind_bl<<|nat_as_tuple_val)) <: NATURAL1
        =>
        bfalse;
thEntire.45:
        a <: NATURAL1*address_space
        =>
        ran (a) <: address_space;
thEntire.46:
        ival: ran(maplet_codomain_indirect)-{null_nat_as_tuple}
        =>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))
        <=
        address_space_size((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival)));
thEntire.47:
        a : ran (maplet_codomain_indirect<+ml_to_be_null*{null_nat_as_tuple}) &
        a /= null_nat_as_tuple
        =>
        a : ran (maplet_codomain_indirect);
thEntire.48:
        a /: dom(ml_to_be_null*{null_nat_as_tuple}) &
        (maplet_codomain_indirect<+ml_to_be_null*{null_nat_as_tuple})(a) /=
null_nat_as_tuple
        =>
        maplet_codomain_indirect (a) /= null_nat_as_tuple;
thEntire.49:
        ml: maplet_codomain &
        maplet_codomain_indirect(ml) /= null_nat_as_tuple
        =>
```

```
        maplet_codomain_real(ml) = null_nat_ptr;
thEntire.50:
        ml : dom (ml_to_be_null*{null_nat_as_tuple}) &
        (maplet_codomain_indirect<+ml_to_be_null*{null_nat_as_tuple})(ml) /=
null_nat_as_tuple
        =>
        bfalse
```

اثبات‌های مربوط به عملگر mapPage :

نیازمندی اثبات شماره ۱:

```
"`Check that the invariant (maplet_codomain_indirect: maplet_codomain -->
nat_as_tuple\/{null_nat_as_tuple}) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_indirect<+{address_space_maplets(pp_as)(faultyPageNo)|-
>new_nat_as_tuple}: maplet_codomain +->
nat_as_tuple\/{new_nat_as_tuple}\/{null_nat_as_tuple}
```

اثبات:

```
ff(0) & dd & pr & ar(thEntire.51,Goal) & ah(ran(maplet_codomain_indirect) <:
dom(nat_as_tuple_val)\/{maplet_codomain_indirect(address_space_maplets(process_address_s
pace(process_pid~(pid)))(faultyPageNo))}) & pr & dd & pr & ar(thEntire.51,Goal) & dd &
ar(thEntire.52,Once) & pr & pr
```

نیازمندی اثبات شماره ۲:

```
"`Check that the invariant (!ival.(ival: ran(maplet_codomain_indirect)-
{null_nat_as_tuple} => (%(pg,as).(pg: NATURAL1 & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: NATURAL1 & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival))))) is preserved by the operation - ref
3.4'"
=>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))((nat_as_tuple_val<+{new_nat_as_tuple|->(pageNo|-
>cas)})(ival))<=address_space_size((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as:
ADDRESS_SPACE | as))((nat_as_tuple_val<+{new_nat_as_tuple|->(pageNo|->cas)})(ival)))
```

اثبات:

```
ff(0) & dd & pr & ah((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))(pageNo|->process_address_space(first(queue_processes(running)))) = pageNo) &
ar(thEntire.53,Once) & pr & pr & dd & ah((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) &
as: ADDRESS_SPACE | as))(pageNo|-
>process_address_space(first(queue_processes(running)))) =
process_address_space(first(queue_processes(running)))) & ar(thEntire.54,Once) & pr & pr
& dd &
ah(pageNo<=address_space_size(process_address_space(first(queue_processes(running))))) &
dd & pr & ar(thEntire.55,Goal) & pr & pr & pr & dd & pr & ar(thEntire.56,Once) &
ar(thEntire.57,Once) & pr & pr & pr
```

نیازمندی اثبات شماره ۳:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_indirect(ml)/=null_nat_as_tuple => maplet_codomain_real(ml) =
null_nat_ptr))) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_real(ml) = null_nat_ptr
```

اثبات:

```
ff(0) & dd & dc(ml = address_space_maplets(pp_as)(faultyPageNo)) & dd & pr & pr
```

نیازمندی اثبات شماره ٤:

```
"`Check that the invariant (maplet_codomain_indirect: maplet_codomain -->
nat_as_tuple\/{null_nat_as_tuple}) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_indirect<+{address_space_maplets(pp_as)(faultyPageNo)|-
>new_nat_as_tuple}: maplet_codomain +->
```

```
nat_as_tuple\/{new_nat_as_tuple}\/{null_nat_as_tuple}
```

اثبات:

```
ff(0) & dd & pr & ar(thEntire.51,Goal) & ar(thEntire.52,Once) & pr & dd &
ar(thEntire.51,Goal) & ar(thEntire.52,Once) & pr & pr
```

نیازمندی اثبات شماره ۵:

```
"`Check that the invariant (!ival.(ival: ran(maplet_codomain_indirect)-
{null_nat_as_tuple} => (%(pg,as).(pg: NATURAL1 & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: NATURAL1 & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival))))) is preserved by the operation - ref
3.4'"
=>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))((nat_as_tuple_val<+{new_nat_as_tuple|->(pageNo|-
>cas)})(ival))<=address_space_size((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as:
ADDRESS_SPACE | as))((nat_as_tuple_val<+{new_nat_as_tuple|->(pageNo|->cas)})(ival)))
```

اثبات:

```
ff(0) & dd & pr & ah((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))(pageNo|->process_address_space(first(queue_processes(running)))) = pageNo) &
ar(thEntire.53,Once) & pr & pr & dd & ah((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) &
as: ADDRESS_SPACE | as))(pageNo|-
>process_address_space(first(queue_processes(running)))) =
process_address_space(first(queue_processes(running)))) & ar(thEntire.54,Once) & pr & pr
& dd & pr & ar(thEntire.55,Goal) & pr & pr & pr & dd & pr & ar(thEntire.56,Once) &
ar(thEntire.57,Once) & pr & pr & pr
```

نیازمندی اثبات شماره ٦:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_indirect(ml)/=null_nat_as_tuple => maplet_codomain_real(ml) =
null_nat_ptr))) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_real(ml) = null_nat_ptr
```

اثبات:

```
ff(0) & dd & dc(ml = address_space_maplets(pp_as)(faultyPageNo)) & dd & pr & pr
```

نیازمندی اثبات شماره ۷:

```
"`Check that the invariant (!ival.(ival: ran(maplet_codomain_indirect)-
{null_nat_as_tuple} => (%(pg,as).(pg: NATURAL1 & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: NATURAL1 & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival))))) is preserved by the operation - ref
3.4'"
=>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: INTEGER & 0<=pg & not(pg
= 0) & as: ADDRESS_SPACE | as))(nat_as_tuple_val(ival)))
```

اثبات:

```
ff(0) & dd & ar(thEntire.56,Once) & ar(thEntire.58,Once) & pr & pr & pr
```

نیازمندی اثبات شماره ۸:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_indirect(ml)/=null_nat_as_tuple => maplet_codomain_real(ml) =
null_nat_ptr))) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_real(ml) = null_nat_ptr
```

اثبات:

```
ff(0) & dd & dc(ml = address_space_maplets(pp_as)(faultyPageNo)) & dd & pr & pr
```

قواعد مورد استفاده در اثبات‌های مربوط به عملگر mapPage

```
thEntire.51:
        a \/ b \/ c == a \/ c \/ b;
thEntire.52:
        a <: b \/ c
        =>
        a <: b \/ c \/ d;
thEntire.53:
        a : address_space &
        p : NATURAL1
        =>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE | pg))(p|->a) =
p;
thEntire.54:
        a : address_space &
        p : NATURAL1
        =>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE | as))(p|->a) =
a;
thEntire.55:
        a : address_space &
        p : NATURAL1
        =>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE | pg))(p|->a) ==
p;
thEntire.56:
        ival: ran(maplet_codomain_indirect) &
        not(ival = null_nat_as_tuple)
        =>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))
        <=
        address_space_size((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival)));
thEntire.57:
        ival /= new_nat_as_tuple &
        ival :
ran(maplet_codomain_indirect<+{address_space_maplets(pp_as)(faultyPageNo)|-
>new_nat_as_tuple})
        =>
        ival : ran(maplet_codomain_indirect);
thEntire.58:
        ival /= null_nat_as_tuple &
        ival :
ran(maplet_codomain_indirect<+{address_space_maplets(pp_as)(faultyPageNo)|-
>maplet_codomain_indirect(address_space_maplets(cas)(pageNo))})
        =>
        ival : ran(maplet_codomain_indirect);
<<USELESS> thEntire.59:
<<USELESS>      ml = address_space_maplets(pp_as)(faultyPageNo) &
<<USELESS>      maplet_codomain_real(address_space_maplets(pp_as)(faultyPageNo)) =
null_nat_ptr &
<<USELESS>      not(maplet_codomain_real(ml) = null_nat_ptr)
<<USELESS>      =>
<<USELESS>      bfalse;
```

اثبات‌های مربوط به عملگر grantPage:

نیازمندی اثبات شماره ۱:

```
"`Check that the invariant (maplet_codomain_real: maplet_codomain -->
nat_ptr\/{null_nat_ptr}) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_real<+{address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real}: maplet_codomain +-
> nat_ptr\/{null_nat_ptr}
```

اثبات:

```
ff(0) & dd & ah({address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real}: maplet_codomain +-
> nat_ptr\/{null_nat_ptr}) & ah(dom({address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real}) <:
```

```
maplet_codomain) & pr & dd & ah(ran({address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real}) <:
nat_ptr\/{null_nat_ptr}) & pr & dd &
ah(address_space_maplets(cas)(pageNo)/=address_space_maplets(pp_as)(faultyPageNo)) &
ah(cas/=pp_as) &
ah(process_address_space(first(queue_processes(running)))/=process_address_space(process
_pid~(pid))) & ah(first(queue_processes(running))/=process_pid~(pid)) &
ar(thEntire.60,Once) & pr & pr & dd & ar(thEntire.61,Once) & pr & pr & dd & ah(cas =
process_address_space(first(queue_processes(running)))) & pr & ah(pp_as =
process_address_space(process_pid~(pid))) & dd & pr & ar(thEntire.62,Once) & pr & dd &
pr & dd & pr
```

نیازمندی اثبات شماره ۲:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_real(ml)/=null_nat_ptr => maplet_codomain_indirect(ml) =
null_nat_as_tuple))) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_indirect(ml) = null_nat_as_tuple
```

اثبات:

```
ff(0) & dd & ah(ml/=address_space_maplets(cas)(pageNo)) & ct & ar(thEntire.63,Once) & pr
& pr & pr & dd & dc(ml = address_space_maplets(pp_as)(faultyPageNo)) & dd &
ar(thEntire.64,Once) & pr & pr & ah(ml/:dom({address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real})) & pr & dd &
ah((maplet_codomain_real<+{address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real})(ml) =
maplet_codomain_real(ml)) & pr & dd & pr
```

نیازمندی اثبات شماره ۳:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_indirect(ml)/=null_nat_as_tuple => maplet_codomain_real(ml) =
null_nat_ptr))) is preserved by the operation - ref 3.4'"
=>
        (maplet_codomain_real<+{address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real})(ml) = null_nat_ptr
```

اثبات:

```
ff(0) & dd & ah(ml/=address_space_maplets(pp_as)(faultyPageNo)) & ct &
ar(thEntire.65,Once) & pr & pr & pr & dd & dc(ml:
dom({address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real})) & dd & pr & pr &
dd & pr
```

نیازمندی اثبات شماره ٤:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_real(ml)/=null_nat_ptr => maplet_codomain_indirect(ml) =
null_nat_as_tuple))) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_indirect(ml) = null_nat_as_tuple
```

اثبات:

```
ff(0) & dd & ah(ml/=address_space_maplets(cas)(pageNo)) & ct & ar(thEntire.66,Once) & pr
& pr & pr & dd & dc(ml: dom({address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real})) & dd & pr & dd &
ar(thEntire.67,Once) & pr & pr
```

نیازمندی اثبات شماره ۵:

```
"`Check that the invariant (maplet_codomain_indirect: maplet_codomain -->
nat_as_tuple\/{null_nat_as_tuple}) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_indirect<+{address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple}: maplet_codomain +->
nat_as_tuple\/{null_nat_as_tuple}
```

اثبات:

```
ff(0) & dd & ah({address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple}: maplet_codomain +->
nat_as_tuple\/{null_nat_as_tuple}) &
ah(dom({address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple}) <: maplet_codomain) &
pr & dd & ah(ran({address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple}) <:
nat_as_tuple\/{null_nat_as_tuple}) & pr & dd &
ah(address_space_maplets(cas)(pageNo)/=address_space_maplets(pp_as)(faultyPageNo)) &
ah(cas/=pp_as) &
ah(process_address_space(first(queue_processes(running)))/=process_address_space(process
_pid~(pid))) & ah(first(queue_processes(running))/=process_pid~(pid)) &
ar(thEntire.60,Once) & pr & pr & dd & ar(thEntire.61,Once) & pr & pr & dd & pr & ah(cas
= process_address_space(first(queue_processes(running)))) & ah(pp_as =
process_address_space(process_pid~(pid))) & dd & ar(thEntire.62,Once) & pr & dd & pr &
dd & pr
```

نیازمندی اثبات شماره ٦:

```
"`Check that the invariant (!ival.(ival: ran(maplet_codomain_indirect)-
{null_nat_as_tuple} => (%(pg,as).(pg: NATURAL1 & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: NATURAL1 & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival))))) is preserved by the operation - ref
3.4'"
=>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: INTEGER & 0<=pg & not(pg
= 0) & as: ADDRESS_SPACE | as))(nat_as_tuple_val(ival)))
```

اثبات:

```
ff(0) & dd & ah(cas_ind: ran(maplet_codomain_indirect)) & pr & dd &
ah(ran(maplet_codomain_indirect<+{address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple}) =
ran(maplet_codomain_indirect)) & ar(thEntire.68,Once) & pr & dd & pr
```

نیازمندی اثبات شماره ۷:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_real(ml)/=null_nat_ptr => maplet_codomain_indirect(ml) =
null_nat_as_tuple))) is preserved by the operation - ref 3.4'"
=>
        (maplet_codomain_indirect<+{address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple})(ml) =
null_nat_as_tuple
```

اثبات:

```
ff(0) & dd & ah(ml/=address_space_maplets(cas)(pageNo)) & ct & ar(thEntire.69,Once) & pr
& pr & pr & dd & ah(ml/=address_space_maplets(pp_as)(faultyPageNo)) & ct &
ar(thEntire.70,Once) & pr & pr & pr & dd &
ah(ml/:dom({address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple})) & pr & dd & pr
```

نیازمندی اثبات شماره ۸:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_indirect(ml)/=null_nat_as_tuple => maplet_codomain_real(ml) =
null_nat_ptr))) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_real(ml) = null_nat_ptr
```

اثبات:

```
ff(0) & dd & ah(ml/=address_space_maplets(cas)(pageNo)) & ct & ar(thEntire.71,Once) & pr
& ah(ml: dom({address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple})) & pr & dd & pr & pr &
dd & dc(ml: dom({address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple})) & dd & ah(ml =
address_space_maplets(pp_as)(faultyPageNo)) & pr & dd &
```

```
ah((maplet_codomain_indirect<+{address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple})(ml) =
maplet_codomain_indirect(ml)) & pr & dd & ar(thEntire.72,Once) & pr & pr
```

قواعد مورد استفاده در اثبات‌های مربوط به عملگر grantPage

```
thEntire.60:
        pid /: process_pid [ran (a)] &
        process_pid : process >-> NATURAL1
        =>
        first(a) /= process_pid~ (pid);
thEntire.61:
        a : process >->> address_space &
        b /= c
        =>
        a (b) /= a (c);
thEntire.62:
        b /= d
        =>
        a(b)(c) /= a(d)(e);
thEntire.63:
        ml = address_space_maplets(cas)(pageNo) &
        ml : dom ({address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real})&
         (maplet_codomain_real<+
                 {address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real})(ml) /=
null_nat_ptr
        =>
        bfalse;
thEntire.64:
        ml : maplet_codomain &
        ml /= address_space_maplets(pp_as)(faultyPageNo) &
        maplet_codomain_real (ml) /= null_nat_ptr
        =>
        maplet_codomain_indirect(ml) = null_nat_as_tuple;
thEntire.65:
        ml = address_space_maplets(pp_as)(faultyPageNo) &
        maplet_codomain_indirect(address_space_maplets(pp_as)(faultyPageNo)) =
null_nat_as_tuple &
        not(maplet_codomain_indirect(ml) = null_nat_as_tuple)
        =>
        bfalse;
thEntire.66:
        ml = address_space_maplets(cas)(pageNo) &
         (maplet_codomain_real<+
                 {address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real}) (ml) =
null_nat_ptr &
        not((maplet_codomain_real<+
                 {address_space_maplets(cas)(pageNo)|-
>null_nat_ptr,address_space_maplets(pp_as)(faultyPageNo)|->cas_real}) (ml) =
null_nat_ptr)
        =>
        bfalse;
thEntire.67:
        ml: maplet_codomain &
        maplet_codomain_real(ml) /= null_nat_ptr
        =>
        maplet_codomain_indirect(ml) = null_nat_as_tuple;
thEntire.68:
        ran (b) <: ran (a)
        =>
        ran (a<+b) = ran (a);
thEntire.69:
        ml = address_space_maplets(cas)(pageNo) &
        maplet_codomain_real (ml) /= null_nat_ptr &
        maplet_codomain_real(address_space_maplets(cas)(pageNo)) = null_nat_ptr
        =>
        bfalse;
thEntire.70:
        ml = address_space_maplets(pp_as)(faultyPageNo) &
        maplet_codomain_real (ml) /= null_nat_ptr &
        maplet_codomain_real(address_space_maplets(pp_as)(faultyPageNo)) = null_nat_ptr
        =>
```

```
        bfalse;
thEntire.71:
        ml = address_space_maplets(cas)(pageNo) &
        (maplet_codomain_indirect<+
                {address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple}) (ml) =
null_nat_as_tuple &
        not((maplet_codomain_indirect<+
                {address_space_maplets(pp_as)(faultyPageNo)|-
>cas_ind,address_space_maplets(cas)(pageNo)|->null_nat_as_tuple}) (ml) =
null_nat_as_tuple)
        =>
        bfalse;
thEntire.72:
        ml: maplet_codomain & maplet_codomain_indirect(ml) /=null_nat_as_tuple
        =>
        maplet_codomain_real(ml) = null_nat_ptr
```

اثبات‌های مربوط به عملگر **forceSchedule**:

نیازمندی اثبات شماره ۱:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)}:
queue +-> iseq(process)
```

اثبات:

```
ff(0) & dd & pr & ah(UNION(qq).(qq: dom(queue_processes) | ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & pr & ar(thEntire.5,Once) & dd & ar(thEntire.5,Once) & dd &
ah(ran(queue_processes(blocked))-{pp} <: UNION(qq).(qq: dom(queue_processes) |
ran(queue_processes(qq)))) & pr & ar(thEntire.73,Once) & pr & dd & ar(thEntire.74,Once)
& pr & pr & pr & pr & pr & dd & pr & pr & pr & ct & ah(INTER(qq).(qq: queue |
ran(queue_processes(qq))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & pr & pr & ar(thEntire.6,Once) & dd &
ah(ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(bloc
ked))) = {}) & ar(thEntire.15,Once) & pr & pr & dd & ar(thEntire.75,Once) & pr & pr & pr
& pr
```

نیازمندی اثبات شماره ۲:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        dom(queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})
= queue
```

اثبات:

```
ff(0) & dd & pr & pr
```

نیازمندی اثبات شماره ۳:

```
"`Check that the invariant
(size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the
operation - ref 3.4'"
=>
        1<=size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})(running))+size((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(ready))
```

اثبات:

```
ff(0) & dd & ah(running/:dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})) & pr & dd & ah(size((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(running)) = size(queue_processes(running))) & pr & dd &
ah(size(queue_processes(running)) = 1) & ar(thEntire.13,Once) & pr & pr & dd & pr
```

نیازمندی اثبات شماره ٤:

```
"`Check that the invariant (size(queue_processes(running))<=1) is preserved by the
operation - ref 3.4'"
=>
        size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})(running))<=1
```

اثبات:

```
ff(0) & dd & ah(running/:dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})) & pr & dd & ah((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(running) = queue_processes(running)) & pr & dd & pr
```

نیازمندی اثبات شماره ٥:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(qq))) = process
```

اثبات:

```
ff(0) & dd & pr & pr & ah(UNION(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(qq))) = ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(ready))\/(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))\/ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) & ar(thEntire.5,Once) & dd &
ah(ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(ready))\/(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))\/ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & ah(ready: dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})) & pr & dd & ah(blocked: dom({blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})) & pr & dd & ah(running/:dom({blocked|-
>blocked_q,ready|->(queue_processes(ready)<-process_pid~(pid))})) & pr & dd & pr &
ar(thEntire.76,Goal) & ar(thEntire.13,Once) & pr & pr & pr & dd & pr & pr &
ah(UNION(qq).(qq: {running,ready,blocked} | ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & ar(thEntire.5,Once) & dd & pr
```

نیازمندی اثبات شماره ٦:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(qq))) = {}
```

اثبات:

```
ff(0) & dd & pr & pr & ah(INTER(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(qq))) = ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(ready))/\(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))/\ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) & ar(thEntire.6,Once) & dd &
ah(ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(ready))/\(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))/\ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & ah(ready: dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})) & pr & dd & ah(blocked: dom({blocked|->blocked_q,ready|-
```

```
>(queue_processes(ready)<-process_pid~(pid)))) & pr & dd & ah(running/:dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-process_pid~(pid))})) & pr & dd & pr & ar(thEntire.77,Goal) & pr & dd & pr & ar(thEntire.15,Once) & pr & ah(queue = {running,ready,blocked}) & ar(thEntire.6,Once)
```

نیازمندی اثبات شماره ۷:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by the operation - ref 3.4'"
=>
        queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)}: queue +-> iseq(process)
```

اثبات:

```
ff(0) & dd & pr & ah(UNION(qq).(qq: dom(queue_processes) | ran(queue_processes(qq))) = ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked)))) & pr & ar(thEntire.5,Once) & dd & ar(thEntire.5,Once) & dd & ah(ran(queue_processes(blocked))-{pp} <: UNION(qq).(qq: dom(queue_processes) | ran(queue_processes(qq)))) & pr & ar(thEntire.73,Once) & pr & dd & ar(thEntire.74,Once) & pr & pr & pr & pr & pr & dd & pr & pr & pr & ct & ah(INTER(qq).(qq: queue | ran(queue_processes(qq))) = ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked)))) & pr & pr & ar(thEntire.6,Once) & dd & ah(ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked))) = {}) & ar(thEntire.15,Once) & pr & pr & dd & ar(thEntire.75,Once) & pr & pr & pr & pr
```

نیازمندی اثبات شماره ۸:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by the operation - ref 3.4'"
=>
        dom(queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)}) = queue
```

اثبات:

```
ff(0) & dd & pr & pr
```

نیازمندی اثبات شماره ۹:

```
"`Check that the invariant (size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the operation - ref 3.4'"
=>
        1<=size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})(running))+size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})(ready))
```

اثبات:

```
ff(0) & dd & ah(running/:dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})) & pr & dd & ah(size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})(running)) = size(queue_processes(running))) & pr & dd & ah(size(queue_processes(running)) = 1) & ar(thEntire.13,Once) & pr & pr & dd & pr
```

نیازمندی اثبات شماره ۱۰:

```
"`Check that the invariant (size(queue_processes(running))<=1) is preserved by the operation - ref 3.4'"
=>
        size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})(running))<=1
```

اثبات:

```
ff(0) & dd & ah(running/:dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})) & pr & dd & ah((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})(running) = queue_processes(running)) & pr & dd & pr
```

نیازمندی اثبات شماره ۱۱:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(qq))) = process
```

اثبات:

```
ff(0) & dd & ah(UNION(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(qq))) = ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(ready))\/(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))\/ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) & ar(thEntire.5,Once) & dd &
pr & pr & ah(UNION(qq).(qq: {running,ready,blocked} | ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & ar(thEntire.5,Once) & dd & pr & ar(thEntire.78,Goal) & ar(thEntire.79,Once)
```

نیازمندی اثبات شماره ۱۲:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(qq))) = {}
```

اثبات:

```
ff(0) & dd & pr & pr & ah(INTER(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(qq))) = ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(ready))/\(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))/\ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) & ar(thEntire.6,Once) & dd &
ah(ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(ready))/\(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))/\ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & ar(thEntire.80,Goal) & ar(thEntire.81,Once) & dd & pr & ar(thEntire.15,Once) & pr
& ah(queue = {running,ready,blocked}) & ar(thEntire.6,Once)
```

نیازمندی اثبات شماره ۱۳:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)}:
queue +-> iseq(process)
```

اثبات:

```
ff(0) & dd & pr & ah(UNION(qq).(qq: dom(queue_processes) | ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & pr & ar(thEntire.5,Once) & dd & ar(thEntire.5,Once) & dd &
ah(ran(queue_processes(blocked))-{pp} <: UNION(qq).(qq: dom(queue_processes) |
ran(queue_processes(qq)))) & pr & ar(thEntire.73,Once) & pr & dd & ar(thEntire.74,Once)
& pr & pr & pr & pr & pr & dd & pr & pr & pr & ct & ah(INTER(qq).(qq: queue |
ran(queue_processes(qq))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & pr & pr & ar(thEntire.6,Once) & dd &
ah(ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(bloc
ked))) = {}) & ar(thEntire.15,Once) & pr & pr & dd & ar(thEntire.75,Once) & pr & pr & pr
& pr
```

نیازمندی اثبات شماره ۱۴:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        dom(queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})
= queue
```

اثبات:

```
ff(0) & dd & pr & pr
```

نیازمندی اثبات شماره ۱۵:

```
"`Check that the invariant
(size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the
operation - ref 3.4'"
=>
        1<=size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})(running))+size((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(ready))
```

اثبات:

```
ff(0) & dd & ah(running/:dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})) & pr & dd & ah(size((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(running)) = size(queue_processes(running))) & pr & dd &
ah(size(queue_processes(running)) = 1) & ar(thEntire.13,Once) & pr & pr & dd & pr
```

نیازمندی اثبات شماره ۱۶:

```
"`Check that the invariant (size(queue_processes(running))<=1) is preserved by the
operation - ref 3.4'"
=>
        size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})(running))<=1
```

اثبات:

```
ff(0) & dd & ah(running/:dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})) & pr & dd & ah((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(running) = queue_processes(running)) & pr & dd & pr
```

نیازمندی اثبات شماره ۱۷:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(qq))) = process
```

اثبات:

```
ff(0) & dd & ah(UNION(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(qq))) = ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(ready))\/(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))\/ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) & ar(thEntire.5,Once) & dd &
pr & pr & ah(UNION(qq).(qq: {running,ready,blocked} | ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & ar(thEntire.5,Once) & dd & pr & ar(thEntire.78,Goal) & ar(thEntire.79,Once)
```

نیازمندی اثبات شماره ۱۸:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{blocked|->blocked_q,ready|-
```

```
>(queue_processes(ready)<-pp)})(qq))) = {}
```

اثبات:

```
ff(0) & dd & pr & pr & ah(INTER(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(qq))) = ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(ready))/\(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))/\ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) & ar(thEntire.6,Once) & dd &
ah(ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(ready))/\(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))/\ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & ar(thEntire.80,Goal) & ar(thEntire.81,Once) & dd & pr & ar(thEntire.15,Once) & pr
& ah(queue = {running,ready,blocked}) & ar(thEntire.6,Once)
```

نیازمندی اثبات شماره ۱۹:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)}:
queue +-> iseq(process)
```

اثبات:

```
ff(0) & dd & pr & ah(UNION(qq).(qq: dom(queue_processes) | ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & pr & ar(thEntire.5,Once) & dd & ar(thEntire.5,Once) & dd &
ah(ran(queue_processes(blocked)-{pp} <: UNION(qq).(qq: dom(queue_processes) |
ran(queue_processes(qq)))) & pr & ar(thEntire.73,Once) & pr & dd & ar(thEntire.74,Once)
& pr & pr & pr & pr & pr & dd & pr & pr & pr & ct & ah(INTER(qq).(qq: queue |
ran(queue_processes(qq))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & pr & pr & ar(thEntire.6,Once) & dd &
ah(ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(bloc
ked))) = {}) & ar(thEntire.15,Once) & pr & pr & dd & ar(thEntire.75,Once) & pr & pr & pr
& pr
```

نیازمندی اثبات شماره ۲۰:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        dom(queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-pp)})
= queue
```

اثبات:

```
ff(0) & dd & ah(UNION(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(qq))) = ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(ready))\/(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))\/ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) & ar(thEntire.5,Once) & dd &
pr & pr & ah(UNION(qq).(qq: {running,ready,blocked} | ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & ar(thEntire.5,Once) & dd & pr & ar(thEntire.78,Goal) & ar(thEntire.79,Once)
```

نیازمندی اثبات شماره ۲۱:

```
"`Check that the invariant
(size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the
operation - ref 3.4'"
=>
        1<=size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
```

```
pp)})(running))+size((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(ready))
```

اثبات:

```
ff(0) & dd & ah(running/:dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})) & pr & dd & ah(size((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(running)) = size(queue_processes(running))) & pr & dd &
ah(size(queue_processes(running)) = 1) & ar(thEntire.13,Once) & pr & pr & dd & pr
```

نیازمندی اثبات شماره ۲۲:

```
"`Check that the invariant (size(queue_processes(running))<=1) is preserved by the
operation - ref 3.4'"
=>
        size((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})(running))<=1
```

اثبات:

```
ff(0) & dd & ah(running/:dom({blocked|->blocked_q,ready|->(queue_processes(ready)<-
pp)})) & pr & dd & ah((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(running) = queue_processes(running)) & pr & dd & pr
```

نیازمندی اثبات شماره ۲۳:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(qq))) = process
```

اثبات:

```
ff(0) & dd & ah(UNION(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(qq))) = ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(ready))\/(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))\/ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) & ar(thEntire.5,Once) & dd &
pr & pr & ah(UNION(qq).(qq: {running,ready,blocked} | ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & ar(thEntire.5,Once) & dd & pr & ar(thEntire.78,Goal) & ar(thEntire.79,Once)
```

نیازمندی اثبات شماره ۲٤:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-pp)})(qq))) = {}
```

اثبات:

```
ff(0) & dd & pr & pr & ah(INTER(qq).(qq: {running,ready,blocked} |
ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(qq))) = ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(ready))/\(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))/\ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) & ar(thEntire.6,Once) & dd &
ah(ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(ready))/\(ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-
process_pid~(pid))})(running))/\ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked))) =
ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(blocked
)))) & ar(thEntire.80,Goal) & ar(thEntire.81,Once) & dd & pr & ar(thEntire.15,Once) & pr
& ah(queue = {running,ready,blocked}) & ar(thEntire.6,Once)
```

قواعد مورد استفاده در اثبات‌های مربوط به عملگر forceSchedule

```
thEntire.73:
        a <: d
        =>
        a <: b \/ (c \/ d);
thEntire.74:
        a : seq (ran(queue_processes(blocked))-{pp}) &
        ran(queue_processes(blocked))-{pp} <: b
        =>
        a : seq (b);
thEntire.75:
        pp : ran(queue_processes(blocked)) &
        pp : ran(queue_processes(ready)) &
        ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_processes(
blocked))) = {}
        =>
        bfalse;
thEntire.76:
        size (a) = 1
        =>
        {first (a)} \/ b \/ (ran (a) \/ c - {first (a)}) == b \/ (ran (a) \/ c);
thEntire.77:
        ({first(a)}\/b/\(ran(a)/\c))-{first(a)}
        ==
        b/\(ran (a)/\c);
thEntire.78:
        ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(ready))
        \/
        (ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(running))
                \/ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked)))
        ==
        (ran (queue_processes(ready)) \/ {process_pid~(pid)} \/ (ran
(queue_processes(running)) \/ ran (queue_processes(blocked)))) - {process_pid~(pid)};
thEntire.79:
        (a \/ {b} \/ (c \/ d)) - {b} = a \/ (c \/ d);
thEntire.80:
        ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(ready))
        /\
        (ran((queue_processes<+{blocked|->blocked_q,ready|->(queue_processes(ready)<-
process_pid~(pid))})(running))
                /\ran((queue_processes<+{blocked|->blocked_q,ready|-
>(queue_processes(ready)<-process_pid~(pid))})(blocked)))
        ==
        (ran(queue_processes(ready)) \/
{process_pid~(pid)})/\(ran(queue_processes(running))/\(ran(queue_processes(blocked)) -
{process_pid~(pid)}));
thEntire.81:
        a \/ {b} /\ (c /\ d - {b}) = a /\ (c /\ d);
thEntire.82:
        pool_of_free_page_directory_tables : FIN (INTEGER) &
        card (pool_of_free_page_directory_tables) > 0 &
        b : pool_of_free_page_directory_tables
        =>
        b : INTEGER;
thEntire.83: <<USELESS>>
        pool_of_free_page_directory_tables : FIN(NATURAL) &
        fpdt : pool_of_free_page_directory_tables &
        card (pool_of_free_page_directory_tables) > 0 &
        not (fpdt >= 0)
        =>
        bfalse;
```

اثبات‌های مربوط به عملگر createProcess:

نیازمندی اثبات شماره ۱:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
```

```
        queue_processes<+{ready|->(queue_processes(ready)<-new_pr)}: queue +->
iseq(process\/{new_pr})
```

اثبات:

```
ff(0) & dd & ah(ran(queue_processes) <: iseq(process\/{new_pr})) &
ah(ran(queue_processes) <: iseq(process)) & pr & dd & ar(thEntire.98,Once) & pr & dd &
ah({ready|->(queue_processes(ready)<-new_pr)}: queue +-> iseq(process\/{new_pr})) & pr &
dd & pr
```

نیازمندی اثبات شماره ۲:

```
"`Check that the invariant (address_space_maplets: address_space -->
seq1(maplet_codomain)) is preserved by the operation - ref 3.4'"
=>
        address_space_maplets<+{new_as|->mapping}: address_space\/{new_as} +->
seq(maplet_codomain\/new_maplets)-{{}}
```

اثبات:

```
ff(0) & dd & pr & ah(ran(address_space_maplets) <: seq(dom(maplet_codomain_indirect))) &
ah(ran(address_space_maplets) <: seq1(dom(maplet_codomain_indirect))) & dd &
ar(thEntire.84,Once) & pr & dd & ah(ran(address_space_maplets) <:
seq1(dom(maplet_codomain_indirect)\/ran(mapping))) & ar(thEntire.85,Once) & pr & dd &
ar(thEntire.84,Once) & pr & pr & pr & ct & ar(thEntire.86,Once) & pr & pr & pr &
ah(mapping: seq(ran(mapping))) & dd & ar(thEntire.87,Once) & pr & pr & ah(size(mapping)
= asSize) & pr & ar(thEntire.88,Once) & pr & pr & pr & dd & pr
```

نیازمندی اثبات شماره ۳:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(qq))) = process\/{new_pr}
```

اثبات:

```
ff(0) & dd & ah(UNION(qq).(qq: queue | ran((queue_processes<+{ready|-
>(queue_processes(ready)<-new_pr)})(qq))) = ran((queue_processes<+{ready|-
>(queue_processes(ready)<-new_pr)})(ready))\/(ran((queue_processes<+{ready|-
>(queue_processes(ready)<-new_pr)})(running))\/ran((queue_processes<+{ready|-
>(queue_processes(ready)<-new_pr)})(blocked)))) & ah(queue = {running,ready,blocked}) &
ar(thEntire.5,Once) & dd & ah(ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(ready))\/(ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(running))\/ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(blocked))) =
{new_pr}\/ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_process
es(blocked)))) & ah(ready: dom({ready|->(queue_processes(ready)<-new_pr)})) & pr & dd &
ah(running/:dom({ready|->(queue_processes(ready)<-new_pr)})) & pr & dd &
ah(blocked/:dom({ready|->(queue_processes(ready)<-new_pr)})) & pr & dd & pr & dd &
ar(thEntire.89,Goal) & pr & pr & pr & pr & ah(UNION(qq).(qq: dom(queue_processes) |
ran(queue_processes(qq))) =
ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_processes(blocked
)))) & pr & ar(thEntire.5,Once) & dd & pr
```

نیازمندی اثبات شماره ٤:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(qq))) = {}
```

اثبات:

```
ff(0) & dd & ah(queue = {running,ready,blocked}) & ah(INTER(qq).(qq:
{running,ready,blocked} | ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(qq))) = ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(ready))/\(ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(running))/\ran((queue_processes<+{ready|->(queue_processes(ready)<-
```

```
new_pr})(blocked)))) & ar(thEntire.6,Once) & dd & ah(ran((queue_processes<+{ready|-
>(queue_processes(ready)<-new_pr})(ready))/\(ran((queue_processes<+{ready|-
>(queue_processes(ready)<-new_pr})(running))/\ran((queue_processes<+{ready|-
>(queue_processes(ready)<-new_pr})(blocked))) =
{new_pr}\/ran(queue_processes(ready))/\(ran(queue_processes(running))/\ran(queue_process
es(blocked)))) & ah(ready: dom({ready|->(queue_processes(ready)<-new_pr)})) & pr & dd &
ah(running/:dom({ready|->(queue_processes(ready)<-new_pr)})) & pr & dd &
ah(blocked/:dom({ready|->(queue_processes(ready)<-new_pr)})) & pr & dd & pr & dd &
ah(ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(ready))/\(ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(running))/\ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(blocked))) = {}) & ah(blocked/:dom({ready|->(queue_processes(ready)<-
new_pr)})) & pr & dd & ah(running/:dom({ready|->(queue_processes(ready)<-new_pr)})) & pr
& dd & ah(ready: dom({ready|->(queue_processes(ready)<-new_pr)})) & pr & dd & pr &
ar(thEntire.15,Once) & pr & ah(queue = {running,ready,blocked}) & ar(thEntire.6,Once) &
dd & pr
```

نیازمندی اثبات شماره ۵:

```
"`Check that the invariant (!as.(as: address_space =>
card(address_space_maplets(as))<=max_pg)) is preserved by the operation - ref 3.4'"
=>
        card((address_space_maplets<+{new_as|->mapping})(as))<=max_pg
```

اثبات:

```
ff(0) & dd & dc(as = new_as) & pr & ar(thEntire.90,Once) & pr & pr & pr & dd & pr
```

نیازمندی اثبات شماره ٦:

```
"`Check that the invariant (!ival.(ival: ran(maplet_codomain_indirect)-
{null_nat_as_tuple} => (%(pg,as).(pg: NATURAL1 & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: NATURAL1 & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival))))) is preserved by the operation - ref
3.4'"
=>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=(address_space_size<+{new_as|->asSize})((%(pg,as).(pg:
INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE | as))(nat_as_tuple_val(ival)))
```

اثبات:

```
ff(0) & dd & ah(nat_as_tuple_val(ival): NATURAL1*address_space) & pr & dd &
ah((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
as))[NATURAL1*address_space] = address_space) & ah((%(pg,as).(pg: INTEGER & 0<=pg &
not(pg = 0) & as: ADDRESS_SPACE | as))[(NATURAL-{0})*address_space] = ran((NATURAL-
{0})*address_space)) & ar(thEntire.91,Once) & dd & ah(ran((NATURAL-{0})*address_space) =
address_space) & ar(thEntire.92,Once) & ar(thEntire.93,Once) & dd & ar(thEntire.94,Once)
& pr & pr & dd & ah((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
as))(nat_as_tuple_val(ival)): address_space) & ar(thEntire.95,Once) & pr & pr & dd & pr
& ar(thEntire.96,Once) & ar(thEntire.97,Once) & pr & pr & pr
```

نیازمندی اثبات شماره ۷:

```
"`Check that the invariant (!as_obj.(as_obj: address_space =>
size(address_space_maplets(as_obj)) = address_space_size(as_obj))) is preserved by the
operation - ref 3.4'"
=>
        size((address_space_maplets<+{new_as|->mapping})(as_obj)) =
(address_space_size<+{new_as|->asSize})(as_obj)
```

اثبات:

```
ff(0) & dd & pr & ar(thEntire.88,Once) & pr & pr & pr & dd & pr
```

نیازمندی اثبات شماره ۸:

```
"`Check that the invariant (!as_obj.(as_obj: address_space =>
address_space_size(as_obj)<=max_pg)) is preserved by the operation - ref 3.4'"
=>
```

```
        (address_space_size<+{new_as|->asSize})(as_obj)<=max_pg
```

اثبات:

```
ff(0) & dd & pr & pr & dd & pr
```

نیازمندی اثبات شماره ۹:

```
"`Check that the invariant
(card(pool_of_free_page_directory_tables)+card(process)>=max_pr) is preserved by the
operation - ref 3.4'"
=>
        max_pr<=card(pool_of_free_page_directory_tables-{fpdt})+card(process\/{new_pr})
```

اثبات:

```
ff(0) & dd & ah(card(process\/{new_pr}) = card(process)+1) & pr & dd &
ah(card(pool_of_free_page_directory_tables-{fpdt}) =
card(pool_of_free_page_directory_tables)-1) & ah(fpdt:
pool_of_free_page_directory_tables) & dd & ar(thEntire.99,Once) & pr & dd &
ar(thEntire.100,Goal) & pr & ar(thEntire.101,Goal) & pr & pr
```

نیازمندی اثبات شماره ۱۰:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_real(ml)/=null_nat_ptr => maplet_codomain_indirect(ml) =
null_nat_as_tuple))) is preserved by the operation - ref 3.4'"
=>
        (maplet_codomain_indirect\/new_maplets*{null_nat_as_tuple})(ml) =
null_nat_as_tuple
```

اثبات:

```
ff(0) & dd & pr & ah(ml/:ran(mapping)) & ct & ar(thEntire.102,Once) & pr & pr & dd &
ah((maplet_codomain_indirect\/ran(mapping)*{null_nat_as_tuple})(ml) =
maplet_codomain_indirect(ml)) & pr & dd & pr & ar(thEntire.104,Once) &
ar(thEntire.103,Once) & pr & pr & dd & ah(not(maplet_codomain_real(ml) = null_nat_ptr))
& ar(thEntire.103,Once) & pr & pr & dd & ar(thEntire.104,Once) & pr & dd & pr & pr & dd
& ah(ml/:ran(mapping)) & ct & ar(thEntire.102,Once) & pr & pr & dd &
ah((maplet_codomain_indirect\/ran(mapping)*{null_nat_as_tuple})(ml) =
maplet_codomain_indirect(ml)) & ar(thEntire.105,Once) & pr & dd & pr &
ar(thEntire.104,Once) & pr & ar(thEntire.103,Once) & pr & pr
```

نیازمندی اثبات شماره ۱۱:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_indirect(ml)/=null_nat_as_tuple => maplet_codomain_real(ml) =
null_nat_ptr))) is preserved by the operation - ref 3.4'"
=>
        (maplet_codomain_real\/new_maplets*{null_nat_ptr})(ml) = null_nat_ptr
```

اثبات:

```
ff(0) & dd & pr & ah(ml/:ran(mapping)) & ct & ar(thEntire.106,Once) & pr & pr & dd &
ah((maplet_codomain_real\/ran(mapping)*{null_nat_ptr})(ml) = maplet_codomain_real(ml)) &
ar(thEntire.107,Once) & pr & dd & pr &
ah(maplet_codomain_indirect(ml)/=null_nat_as_tuple) & ar(thEntire.108,Once) & pr & pr &
dd & ar(thEntire.109,Once) & pr & dd & ar(thEntire.109,Once) & ar(thEntire.108,Once) &
pr & pr & dd & ah(ml/:ran(mapping)) & ct & ar(thEntire.106,Once) & pr & pr & dd &
ah((maplet_codomain_real\/ran(mapping)*{null_nat_ptr})(ml) = maplet_codomain_real(ml)) &
ar(thEntire.107,Once) & pr & dd & pr &
ah(maplet_codomain_indirect(ml)/=null_nat_as_tuple) & ar(thEntire.108,Once) & pr & pr &
dd & ar(thEntire.109,Once) & pr & dd & ar(thEntire.109,Once) & ar(thEntire.108,Once) &
pr & pr
```

قواعد مورد استفاده در اثبات‌های مربوط به عملگر createProcess

```
thEntire.84:
        a <: seq1 (b)
        =>
```

```
        a <: seq (b);
thEntire.85:
        ran(address_space_maplets) <: seq1(dom(maplet_codomain_indirect))
        =>
        ran(address_space_maplets) <: seq1(dom(maplet_codomain_indirect)\/ran(mapping));
thEntire.86:
        ran(address_space_maplets) <: seq(maplet_codomain)-{{}} &
        {}: ran(address_space_maplets)
        =>
        bfalse;
thEntire.87:
        a : perm (n)
        =>
        a : seq (m\/n);
thEntire.88:
        a : seq (new_maplets) &
        card (new_maplets) = asSize &
        ran (a) = new_maplets
        =>
        size (a) = card (ran (a));
thEntire.89:
        UNION(qq).(qq: queue | ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(qq)))
        =
        {new_pr}\/ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_
processes(blocked)))
        =>
        UNION(qq).(qq: queue | ran((queue_processes<+{ready|->(queue_processes(ready)<-
new_pr)})(qq)))
        ==
        {new_pr}\/ran(queue_processes(ready))\/(ran(queue_processes(running))\/ran(queue_
processes(blocked)));
thEntire.90:
        ran (s) = new_maplets &
        card (new_maplets) = asSize &
        asSize <= m
        =>
        size (s) <= m;
thEntire.91:
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE | as))[a*b] =
ran(a*b);
thEntire.92:
        a /= {}
        =>
        ran (a * b) = b;
thEntire.93:
        NATURAL1 /= {};
thEntire.94:
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE | as))[a*b] =
ran (a*b) &
        ran (a*b) = b
        =>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE | as))[a*b] = b;
thEntire.95:
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
as))[NATURAL1*address_space] = address_space &
        nat_as_tuple_val(ival) : NATURAL1*address_space
        =>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
as))(nat_as_tuple_val(ival)) : address_space;
thEntire.96:
        ival : ran(maplet_codomain_indirect) &
        ival /= null_nat_as_tuple
        =>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))
        <=
        address_space_size((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival)));
thEntire.97:
        ival : ran(maplet_codomain_indirect\/new_maplets*{null_nat_as_tuple}) &
        ival /= null_nat_as_tuple
        =>
        ival : ran(maplet_codomain_indirect);
thEntire.98:
        a <: iseq (p)
        =>
```

```
        a <: iseq (p \/ {q});
thEntire.99:
        a : s
        =>
        card (s - {a}) = card (s) - 1;
thEntire.100:
        card (p\/{q}) = card (p) + 1
        =>
        card (p\/{q}) == card (p) + 1;
thEntire.101:
        card (p-{q}) = card (p) - 1
        =>
        card (p-{q}) == card (p) - 1;
thEntire.102:
        ml : ran(mapping) &
        (maplet_codomain_real\/ran(mapping)*{null_nat_ptr})(ml) /= null_nat_ptr
        =>
        bfalse;
thEntire.103:
        (maplet_codomain_real\/ran(mapping)*{null_nat_ptr})(ml) /= null_nat_ptr &
        ml /: ran(mapping)
        =>
        maplet_codomain_real (ml) /= null_nat_ptr;
thEntire.104:
        maplet_codomain_real(ml)/=null_nat_ptr
        =>
        maplet_codomain_indirect(ml) = null_nat_as_tuple;
thEntire.105:
        ml /: ran(mapping)
        =>
        (maplet_codomain_indirect\/ran(mapping)*{null_nat_as_tuple})(ml) =
maplet_codomain_indirect (ml);
thEntire.106:
        ml : ran(mapping) &
        (maplet_codomain_indirect\/ran(mapping)*{null_nat_as_tuple})(ml) /=
null_nat_as_tuple
        =>
        bfalse;
thEntire.107:
        ml /: ran(mapping)
        =>
        (maplet_codomain_real\/ran(mapping)*{null_nat_ptr})(ml) = maplet_codomain_real
(ml);
thEntire.108:
        (maplet_codomain_indirect\/ran(mapping)*{null_nat_as_tuple})(ml) /=
null_nat_as_tuple &
        ml /: ran(mapping)
        =>
        maplet_codomain_indirect (ml) /= null_nat_as_tuple;
thEntire.109:
        maplet_codomain_indirect(ml)/=null_nat_as_tuple
        =>
        maplet_codomain_real(ml) = null_nat_ptr
```

اثبات‌های مربوط به عملگر **abortProcess**:

نیازمندی اثبات شماره ۱:

```
"`Check that the invariant (nat_as_tuple: FIN(NAT_AS_TUPLE)) is preserved by the
operation - ref 3.4'"
=>
        nat_as_tuple-(nat_as_tuple_to_be_del\/err_tuples_to_be_del): FIN(NAT_AS_TUPLE)
```

اثبات:

```
ff(0) & dd & ah(nat_as_tuple: FIN(NAT_AS_TUPLE)) & dd & ah(nat_as_tuple_to_be_del <:
ran(maplet_codomain_indirect)-{null_nat_as_tuple}) & pr & dd & ah(err_tuples_to_be_del
<: ran(maplet_codomain_indirect)-{null_nat_as_tuple}) & ar(thEntire.111,Once) & dd &
ah(nat_as_tuple_to_be_del\/err_tuples_to_be_del <: ran(maplet_codomain_indirect)-
{null_nat_as_tuple}) & pr & dd & ah(ran(maplet_codomain_indirect) <: NAT_AS_TUPLE) &
ar(thEntire.112,Once) & dd & ar(thEntire.113,Once) & pr & pr & pr & pr
```

نیازمندی اثبات شماره ۲:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        queue_processes<+{ready|->ready_q,running|->running_q,blocked|->blocked_q}:
queue +-> iseq(process-bl)
```

اثبات:

```
ff(0) & dd & ah(queue_processes: queue +-> iseq(process-bl)) & ar(thEntire.114,Once) &
pr & dd & ah({ready|->ready_q,running|->running_q,blocked|->blocked_q}: queue +->
iseq(process-bl)) & ah(dom({ready|->ready_q,running|->running_q,blocked|->blocked_q}) =
queue) & pr & dd & ah(ready/=running & running/=blocked & ready/=blocked) & pr & pr & pr
& dd & ah(ready_q: perm(ran(queue_processes(ready))-bl)) & pr & dd & ah(running_q:
perm(ran(queue_processes(running))-bl)) & pr & dd & ah(blocked_q:
perm(ran(queue_processes(blocked))-bl)) & pr & dd & ar(thEntire.115,Once) & pr & pr & pr
& pr & pr & pr & dd & pr
```

نیازمندی اثبات شماره ۳:

```
"`Check that the invariant (queue_processes: queue --> iseq(process)) is preserved by
the operation - ref 3.4'"
=>
        dom(queue_processes<+{ready|->ready_q,running|->running_q,blocked|->blocked_q})
= queue
```

اثبات:

```
ff(0) & dd & pr & pr & pr
```

نیازمندی اثبات شماره ٤:

```
"`Check that the invariant (address_space_maplets: address_space -->
seq1(maplet_codomain)) is preserved by the operation - ref 3.4'"
=>
        as_obj_to_be_del<<|address_space_maplets: address_space-as_obj_to_be_del +->
seq(maplet_codomain-mpc_obj_to_be_del)-{{}}
```

اثبات:

```
ff(0) & dd & ah(address_space_maplets: address_space +-> seq1(maplet_codomain)) & dd &
ah(dom(as_obj_to_be_del<<|address_space_maplets) <: address_space-as_obj_to_be_del) & pr
& dd & ah(ran(as_obj_to_be_del<<|address_space_maplets) <: seq(maplet_codomain-
mpc_obj_to_be_del)-{{}}) & ar(thEntire.119,Once) & pr & dd & pr
```

نیازمندی اثبات شماره ٥:

```
"`Check that the invariant (maplet_codomain_real: maplet_codomain -->
nat_ptr\/{null_nat_ptr}) is preserved by the operation - ref 3.4'"
=>
        mpc_obj_to_be_del<<|maplet_codomain_real<+res_to_be_return: maplet_codomain-
mpc_obj_to_be_del +-> nat_ptr\/{null_nat_ptr}
```

اثبات:

```
ff(0) & dd & pr & ah(ran(root_server_phi_mpc) <: maplet_codomain) & pr & dd & pr &
ar(thEntire.118,Once) & pr & ah(ran(nat_ptr_seq) <: nat_ptr) & pr & pr & dd & pr
```

نیازمندی اثبات شماره ٦:

```
"`Check that the invariant (maplet_codomain_real: maplet_codomain -->
nat_ptr\/{null_nat_ptr}) is preserved by the operation - ref 3.4'"
=>
        dom(mpc_obj_to_be_del<<|maplet_codomain_real<+res_to_be_return) =
maplet_codomain-mpc_obj_to_be_del
```

اثبات:

```
ff(0) & dd & pr & ah(ran(root_server_phi_mpc) <: maplet_codomain) & pr & dd & pr &
ar(thEntire.117,Once)
```

نیازمندی اثبات شماره ۷:

```
"`Check that the invariant (maplet_codomain_indirect: maplet_codomain -->
nat_as_tuple\/{null_nat_as_tuple}) is preserved by the operation - ref 3.4'"
=>

mpc_obj_to_be_del<<|maplet_codomain_indirect<+mpc_to_be_adjust*{null_nat_as_tuple}:
maplet_codomain-mpc_obj_to_be_del +-> nat_as_tuple-
(nat_as_tuple_to_be_del\/err_tuples_to_be_del)\/{null_nat_as_tuple}
```

اثبات:

```
ff(0) & dd & ah(dom(mpc_obj_to_be_del<<|maplet_codomain_indirect) <: maplet_codomain-
mpc_obj_to_be_del) & pr & dd &
ah(dom(mpc_to_be_adjust*{null_nat_as_tuple})/\mpc_obj_to_be_del = {}) &
ar(thEntire.123,Once) & dd & ah(maplet_codomain_indirect: maplet_codomain +->
nat_as_tuple\/{null_nat_as_tuple}) & dd & ar(thEntire.134,Once) & pr & pr
```

نیازمندی اثبات شماره ۸:

```
"`Check that the invariant (maplet_codomain_indirect: maplet_codomain -->
nat_as_tuple\/{null_nat_as_tuple}) is preserved by the operation - ref 3.4'"
=>

dom(mpc_obj_to_be_del<<|maplet_codomain_indirect<+mpc_to_be_adjust*{null_nat_as_tuple})
= maplet_codomain-mpc_obj_to_be_del
```

اثبات:

```
ff(0) & dd & ah(dom(mpc_obj_to_be_del<<|maplet_codomain_indirect) =
dom(maplet_codomain_indirect)-mpc_obj_to_be_del) & pr & dd & ah(mpc_to_be_adjust <:
dom(maplet_codomain_indirect)) & ar(thEntire.120,Once) & ar(thEntire.121,Once) & pr & dd
& ah(mpc_to_be_adjust <: maplet_codomain) & pr & dd &
ah(dom(mpc_to_be_adjust*{null_nat_as_tuple}) <: maplet_codomain) & pr & dd &
ar(thEntire.122,Once) & ar(thEntire.123,Once) & pr
```

نیازمندی اثبات شماره ۹:

```
"`Check that the invariant (nat_as_tuple_val: nat_as_tuple --> NATURAL1*address_space)
is preserved by the operation - ref 3.4'"
=>
        nat_as_tuple_to_be_del\/err_tuples_to_be_del<<|nat_as_tuple_val: nat_as_tuple-
(nat_as_tuple_to_be_del\/err_tuples_to_be_del) +-> (NATURAL-{0})*(address_space-
as_obj_to_be_del)
```

اثبات:

```
ff(0) & dd & ah(dom(nat_as_tuple_to_be_del\/err_tuples_to_be_del<<|nat_as_tuple_val) <:
nat_as_tuple-(nat_as_tuple_to_be_del\/err_tuples_to_be_del)) & pr & dd &
ah(nat_as_tuple_val: nat_as_tuple +-> NATURAL1*address_space) & dd &
ah(ran(nat_as_tuple_to_be_del\/err_tuples_to_be_del<<|nat_as_tuple_val) <:
NATURAL1*(address_space-as_obj_to_be_del)) & ar(thEntire.124,Once) & dd & pr
```

نیازمندی اثبات شماره ۱۰:

```
"`Check that the invariant (root_server: process) is preserved by the operation - ref
3.4'"
=>
        not(root_server: bl)
```

اثبات:

```
ff(0) & dd & ct & ah(pid>1) & ar(thEntire.125,Once) & pr & pr & pr & dd &
ah(process_pid~(pid)/=root_server) & ar(thEntire.127,Once) & pr & ar(thEntire.126,Once)
& dd & ah(root_server/:process_parent_pid~[{pid}]) & ar(thEntire.128,Once) & pr &
ar(thEntire.126,Once) & dd & ah(root_server/:process_pager_pid~[{pid}]) &
ar(thEntire.129,Once) & pr & ar(thEntire.126,Once) & dd &
```

```
ah(root_server/:process_exman_pid~[{pid}]) & ar(thEntire.130,Once) & pr &
ar(thEntire.126,Once) & dd & ar(thEntire.131,Once) & pr & pr & pr & pr & pr & pr
```

نیازمندی اثبات شماره ۱۱:

```
"`Check that the invariant
(size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the
operation - ref 3.4'"
=>
        1<=size((queue_processes<+{ready|->ready_q,running|->running_q,blocked|-
>blocked_q})(running))+size((queue_processes<+{ready|->ready_q,running|-
>running_q,blocked|->blocked_q})(ready))
```

اثبات:

```
ff(0) & dd & ah(running: dom({ready|->ready_q,running|->running_q,blocked|->blocked_q}))
& pr & dd & ah(ready: dom({ready|->ready_q,running|->running_q,blocked|->blocked_q})) &
pr & dd & pr & dc(pp: ran(queue_processes(blocked))) & dd & ar(thEntire.135,Once) & pr &
dd & ar(thEntire.136,Once) & pr & pr
```

نیازمندی اثبات شماره ۱۲:

```
"`Check that the invariant (size(queue_processes(running))<=1) is preserved by the
operation - ref 3.4'"
=>
        size((queue_processes<+{ready|->ready_q,running|->running_q,blocked|-
>blocked_q})(running))<=1
```

اثبات:

```
ff(0) & dd & ah(running: dom({ready|->ready_q,running|->running_q,blocked|->blocked_q}))
& pr & dd & pr & ar(thEntire.110,Once) & pr & pr
```

نیازمندی اثبات شماره ۱۳:

```
"`Check that the invariant (UNION(qq).(qq: queue | ran(queue_processes(qq))) = process)
is preserved by the operation - ref 3.4'"
=>
        UNION(qq).(qq: queue | ran((queue_processes<+{ready|->ready_q,running|-
>running_q,blocked|->blocked_q})(qq))) = process-bl
```

اثبات:

```
ff(0) & dd & ar(thEntire.137,Once) & ah(queue = {running,ready,blocked}) & pr & pr & pr
& pr
```

نیازمندی اثبات شماره ۱٤:

```
"`Check that the invariant (INTER(qq).(qq: queue | ran(queue_processes(qq))) = {}) is
preserved by the operation - ref 3.4'"
=>
        INTER(qq).(qq: queue | ran((queue_processes<+{ready|->ready_q,running|-
>running_q,blocked|->blocked_q})(qq))) = {}
```

اثبات:

```
ff(0) & dd & ar(thEntire.138,Once) & pr & pr & pr & pr
```

نیازمندی اثبات شماره ۱۵:

```
"`Check that the invariant (!ival.(ival: ran(maplet_codomain_indirect)-
{null_nat_as_tuple} => (%(pg,as).(pg: NATURAL1 & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))<=address_space_size((%(pg,as).(pg: NATURAL1 & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival))))) is preserved by the operation - ref
3.4'"
=>
        (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))((nat_as_tuple_to_be_del\/err_tuples_to_be_del<<|nat_as_tuple_val)(ival))<=(as_obj_t
```

```
o_be_del<<|address_space_size)((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as:
ADDRESS_SPACE |
as))((nat_as_tuple_to_be_del\/err_tuples_to_be_del<<|nat_as_tuple_val)(ival)))
```

اثبات:

```
ff(0) & dd & pr & ar(thEntire.139,Once)
```

نیازمندی اثبات شماره ۱۶:

```
"`Check that the invariant (pool_of_free_page_directory_tables: FIN(NAT)) is preserved
by the operation - ref 3.4'"
=>
        pool_of_free_page_directory_tables\/pdt_to_be_free: FIN(NAT)
```

اثبات:

```
ff(0) & dd & ah(pdt_to_be_free: FIN(NAT)) & ar(thEntire.140,Once) & pr & dd & pr
```

نیازمندی اثبات شماره ۱۷:

```
"`Check that the invariant
(card(pool_of_free_page_directory_tables)+card(process)>=max_pr) is preserved by the
operation - ref 3.4'"
=>
        max_pr<=card(pool_of_free_page_directory_tables\/pdt_to_be_free)+card(process-
bl)
```

اثبات:

```
ff(0) & dd & ah(card(pdt_to_be_free) = card(bl)) & ar(thEntire.141,Once) & dd &
ar(thEntire.142,Once) & pr & pr
```

نیازمندی اثبات شماره ۱۸:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_real(ml)/=null_nat_ptr => maplet_codomain_indirect(ml) =
null_nat_as_tuple))) is preserved by the operation - ref 3.4'"
=>

(mpc_obj_to_be_del<<|maplet_codomain_indirect<+mpc_to_be_adjust*{null_nat_as_tuple})(ml)
= null_nat_as_tuple
```

اثبات:

```
ff(0) & dd & dc(ml: dom(mpc_to_be_adjust*{null_nat_as_tuple})) & dd & pr & dd &
ah(ml/:mpc_obj_to_be_del) & dd & ar(thEntire.143,Goal) & pr & pr & ar(thEntire.144,Once)
& pr & pr & pr
```

نیازمندی اثبات شماره ۱۹:

```
"`Check that the invariant (!ml.(ml: maplet_codomain =>
(maplet_codomain_indirect(ml)/=null_nat_as_tuple => maplet_codomain_real(ml) =
null_nat_ptr))) is preserved by the operation - ref 3.4'"
=>
        (mpc_obj_to_be_del<<|maplet_codomain_real<+res_to_be_return)(ml) = null_nat_ptr
```

اثبات:

```
ff(0) & dd & ah(ml/:dom(res_to_be_return)) & ar(thEntire.145,Once) & pr & dd &
ah(ml/:mpc_obj_to_be_del) & dd & ar(thEntire.146,Goal) & pr & pr & ar(thEntire.147,Once)
& pr & pr & pr
```

نیازمندی اثبات شماره ۲۰:

```
"`Check that the invariant (maplet_codomain_indirect: maplet_codomain -->
nat_as_tuple\/{null_nat_as_tuple}) is preserved by the operation - ref 3.4'"
=>
```

```
mpc_obj_to_be_del<<|maplet_codomain_indirect<+mpc_to_be_adjust*{null_nat_as_tuple}:
maplet_codomain +-> nat_as_tuple-
(nat_as_tuple_to_be_del\/err_tuples_to_be_del)\/{null_nat_as_tuple}
```

اثبات:

```
ff(0) & dd & ah(dom(mpc_obj_to_be_del<<|maplet_codomain_indirect) <:
dom(maplet_codomain_indirect)-mpc_obj_to_be_del) & pr & dd &
ah(dom(maplet_codomain_indirect)-mpc_obj_to_be_del <: maplet_codomain) & pr & dd &
ah(dom(mpc_obj_to_be_del<<|maplet_codomain_indirect) <: maplet_codomain) & pr & dd &
ah(maplet_codomain_indirect: maplet_codomain +-> nat_as_tuple\/{null_nat_as_tuple}) & dd
& ah(dom(mpc_to_be_adjust*{null_nat_as_tuple})/\mpc_obj_to_be_del = {}) &
ar(thEntire.123,Once) & dd & ar(thEntire.148,Once) & pr & pr & pr
```

نیازمندی اثبات شماره ۲۱:

```
"`Check that the invariant (maplet_codomain_real: maplet_codomain -->
nat_ptr\/{null_nat_ptr}) is preserved by the operation - ref 3.4'"
=>
        maplet_codomain_real: maplet_codomain-mpc_obj_to_be_del +->
nat_ptr\/{null_nat_ptr}
```

اثبات:

```
ff(0) & dd & ah(dom(maplet_codomain_real) <: maplet_codomain-mpc_obj_to_be_del) &
ar(thEntire.149,Once) & dd & ah(maplet_codomain_real: maplet_codomain +->
nat_ptr\/{null_nat_ptr}) & dd & ar(thEntire.150,Once) & pr & pr
```

نیازمندی اثبات شماره ۲۲:

```
"`Check that the invariant
(size(queue_processes(running))+size(queue_processes(ready))>0) is preserved by the
operation - ref 3.4'"
=>
        1<=size((queue_processes<+{ready|->ready_q,running|->running_q,blocked|-
>blocked_q})(running))+size((queue_processes<+{ready|->ready_q,running|-
>running_q,blocked|->blocked_q})(ready))
```

اثبات:

```
ff(0) & dd & ah(running: dom({ready|->ready_q,running|->running_q,blocked|->blocked_q}))
& pr & dd & ah(ready: dom({ready|->ready_q,running|->running_q,blocked|->blocked_q})) &
pr & dd & pr & ar(thEntire.135,Once) & pr
```

قواعد مورد استفاده در اثبات‌های مربوط به عملگر abortProcess

```
thEntire.110:
        s : seq (ran(queue_processes(running))-bl) &
        size (queue_processes(running)) <= 1
        =>
        size (s) <= 1;
thEntire.111:
        err_tuples_to_be_del <: ran(maplet_codomain_indirect)-{null_nat_as_tuple}; /*we
can infer its type syntactically*/
thEntire.112:
        ran(maplet_codomain_indirect) <: NAT_AS_TUPLE;
thEntire.113:
        nat_as_tuple : FIN(NAT_AS_TUPLE) &
        nat_as_tuple_to_be_del <: ran (maplet_codomain_indirect) - {null_nat_as_tuple} &
        err_tuples_to_be_del <: ran (maplet_codomain_indirect) - {null_nat_as_tuple} &
        ran (maplet_codomain_indirect) : FIN (NAT_AS_TUPLE)
        =>
        nat_as_tuple-(nat_as_tuple_to_be_del\/err_tuples_to_be_del) : FIN(NAT_AS_TUPLE);
thEntire.114:
        queue_processes: queue +-> iseq(process)
        =>
        queue_processes: queue +-> iseq(process-bl);
thEntire.115:
        running_q : perm (ran(queue_processes(running))-bl) &
        ran(queue_processes(running))-bl <: process-bl &
```

```
        ready_q : perm (ran(queue_processes(ready))-bl) &
        ran(queue_processes(ready))-bl <: process-bl &
        blocked_q : perm (ran(queue_processes(blocked))-bl) &
        ran(queue_processes(blocked))-bl <: process-bl
        =>
        {ready|->ready_q,running|->running_q,blocked|->blocked_q}: queue +->
iseq(process-bl);
thEntire.116:
        dom (address_space_maplets) <: address_space
        =>
        dom(dom(as_obj_to_be_del<<|address_space_maplets) <: address_space-
as_obj_to_be_del);
thEntire.117:
        dom(res_to_be_return)/\mpc_obj_to_be_del = {};
thEntire.118:
        root_server_phi_mpc[1..size(nat_ptr_seq)]/\mpc_obj_to_be_del = {};
thEntire.119:
        mpc_obj_to_be_del = UNION(mpc_s).(mpc_s: address_space_maplets[as_obj_to_be_del]
| ran(mpc_s))
        =>
        ran(as_obj_to_be_del<<|address_space_maplets) <: seq(maplet_codomain-
mpc_obj_to_be_del)-{{}};
thEntire.120:
        mpc_to_be_adjust <: ran (maplet_codomain_indirect~)
        =>
        mpc_to_be_adjust <: dom (maplet_codomain_indirect);
thEntire.121:
        mpc_to_be_adjust = maplet_codomain_indirect~ [err_tuples_to_be_del]
        =>
        mpc_to_be_adjust <: ran (maplet_codomain_indirect~);
thEntire.122:
        dom (mpc_to_be_adjust*{null_nat_as_tuple}) /\ mpc_obj_to_be_del = {} &
        dom (mpc_obj_to_be_del<<|maplet_codomain_indirect) = maplet_codomain-
mpc_obj_to_be_del
        =>
        dom(mpc_obj_to_be_del<<|maplet_codomain_indirect<+mpc_to_be_adjust*{null_nat_as_t
uple}) = maplet_codomain-mpc_obj_to_be_del;
thEntire.123:
        dom (mpc_to_be_adjust*{null_nat_as_tuple}) /\ mpc_obj_to_be_del = {};
thEntire.124:
        ran(nat_as_tuple_to_be_del\/err_tuples_to_be_del<<|nat_as_tuple_val) <: (NATURAL-
{0})*(address_space-as_obj_to_be_del);
thEntire.125:
        process_pid : process >-> NATURAL1 &
        pid /= process_pid (root_server) &
        pid : NATURAL1
        =>
        pid > 1;
thEntire.126:
        process_pid (root_server) = 1;
thEntire.127:
        pid > 1 &
        process_pid (root_server) = 1
        =>
        process_pid~ (pid) /= root_server;
thEntire.128:
        pid > 1 &
        process_pid (root_server) = 1
        =>
        root_server /: process_parent_pid~ [{pid}];
thEntire.129:
        pid > 1 &
        process_pid (root_server) = 1
        =>
        root_server /: process_pager_pid~ [{pid}];
thEntire.130:
        pid > 1 &
        process_pid (root_server) = 1
        =>
        root_server /: process_exman_pid~ [{pid}];
thEntire.131:
        process_pid~ (pid) /= root_server &
        root_server /: process_parent_pid~ [{pid}] &
        root_server /: process_pager_pid~ [{pid}] &
        root_server /: process_exman_pid~ [{pid}] &
        bl =
process_parent_pid~[{pid}]\/process_pager_pid~[{pid}]\/process_exman_pid~[{pid}]\/{proce
```

```
ss_pid~ (pid)} &
       root_server : bl
       =>
       bfalse;
thEntire.132:
       dom(mpc_to_be_adjust*{null_nat_as_tuple})/\mpc_obj_to_be_del = {}
       =>
       mpc_obj_to_be_del/\dom(mpc_to_be_adjust*{null_nat_as_tuple}) = {};
thEntire.133:
       a <: b
       =>
       a <: b \/ c \/ d;
thEntire.134:
       dom(mpc_to_be_adjust*{null_nat_as_tuple})/\mpc_obj_to_be_del = {} &
       dom (mpc_obj_to_be_del<<|maplet_codomain_indirect) <: maplet_codomain-
mpc_obj_to_be_del
       =>
       mpc_obj_to_be_del<<|maplet_codomain_indirect<+mpc_to_be_adjust*{null_nat_as_tuple
} :
       maplet_codomain-mpc_obj_to_be_del +-> nat_as_tuple-
(nat_as_tuple_to_be_del\/err_tuples_to_be_del)\/{null_nat_as_tuple};
thEntire.135:
       pp : ran (queue_processes (blocked))
       =>
       0 <= -1 + size (running_q) + size (ready_q);
thEntire.136:
       pp /: ran (queue_processes (blocked)) &
       card ((ran (queue_processes (running)) \/ ran (queue_processes (ready))) - bl) >
0
       =>
       0 <= -1 + size (running_q) + size (ready_q);
thEntire.137:
       UNION(qq).(qq: queue | ran((queue_processes)(qq))) = process &
       ready_q : perm (ran (queue_processes (ready)) - bl) &
       running_q : perm (ran (queue_processes (running)) - bl) &
       blocked_q : perm (ran (queue_processes (blocked)) - bl)
       =>
       UNION(qq).(qq: queue | ran((queue_processes<+{ready|->ready_q,running|-
>running_q,blocked|->blocked_q})(qq))) = process-bl;
thEntire.138:
       INTER(qq).(qq: queue | ran((queue_processes)(qq))) = {} &
       ready_q : perm (ran (queue_processes (ready)) - bl) &
       running_q : perm (ran (queue_processes (running)) - bl) &
       blocked_q : perm (ran (queue_processes (blocked)) - bl)
       =>
       INTER(qq).(qq: queue | ran((queue_processes<+{ready|->ready_q,running|-
>running_q,blocked|->blocked_q})(qq))) = {};
thEntire.139:
       (%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as: ADDRESS_SPACE |
pg))(nat_as_tuple_val(ival))
       <=
       address_space_size((%(pg,as).(pg: INTEGER & 0<=pg & not(pg = 0) & as:
ADDRESS_SPACE | as))(nat_as_tuple_val(ival)));
thEntire.140:
       pdt_to_be_free = UNION(pr).(pr: bl | {process_system_status(pr)(CR3)})
       =>
       pdt_to_be_free : FIN (NAT);
thEntire.141:
       card(pdt_to_be_free) = card(bl);
thEntire.142:
       card(pdt_to_be_free) = card(bl) &
       max_pr <= card(pool_of_free_page_directory_tables) + card (process)
       =>
       max_pr <= card(pool_of_free_page_directory_tables \/ pdt_to_be_free) + card
(process-bl);
thEntire.143:
       ml /: dom (mpc_to_be_adjust*{null_nat_as_tuple}) &
       ml /: mpc_obj_to_be_del
       =>
       (mpc_obj_to_be_del<<|maplet_codomain_indirect<+mpc_to_be_adjust*{null_nat_as_tupl
e})(ml) == maplet_codomain_indirect (ml);
thEntire.144:
       not((mpc_obj_to_be_del<<|maplet_codomain_real<+res_to_be_return)(ml) =
null_nat_ptr) &
       ml /: dom (mpc_to_be_adjust*{null_nat_as_tuple}) &
       ml /: mpc_obj_to_be_del
       =>
```

```
        maplet_codomain_indirect(ml) = null_nat_as_tuple;
thEntire.145:
        ml : maplet_codomain
        =>
        ml /: dom (res_to_be_return);
thEntire.146:
        ml /: mpc_obj_to_be_del &
        ml /: dom (res_to_be_return)
        =>
        (mpc_obj_to_be_del<<|maplet_codomain_real<+res_to_be_return)(ml) ==
maplet_codomain_real (ml);
thEntire.147:
        ml /: mpc_obj_to_be_del &
        ml /: dom (res_to_be_return) &
        (mpc_obj_to_be_del<<|maplet_codomain_indirect<+mpc_to_be_adjust*{null_nat_as_tupl
e})(ml) /= null_nat_as_tuple
        =>
        maplet_codomain_real(ml) = null_nat_ptr;
thEntire.148:
        maplet_codomain_indirect : maplet_codomain +-> nat_as_tuple \/
{null_nat_as_tuple} &
        dom (mpc_obj_to_be_del<<|maplet_codomain_indirect) <: maplet_codomain &
        dom (mpc_to_be_adjust*{null_nat_as_tuple}) /\ mpc_obj_to_be_del = {}
        =>
        mpc_obj_to_be_del<<|maplet_codomain_indirect<+mpc_to_be_adjust*{null_nat_as_tuple
}
        : maplet_codomain +-> nat_as_tuple-
(nat_as_tuple_to_be_del\/err_tuples_to_be_del)\/{null_nat_as_tuple};
thEntire.149:
        dom(maplet_codomain_real) <: maplet_codomain-mpc_obj_to_be_del;
thEntire.150:
        dom(maplet_codomain_real) <: maplet_codomain-mpc_obj_to_be_del &
        maplet_codomain_real: maplet_codomain +-> nat_ptr\/{null_nat_ptr}
        =>
        maplet_codomain_real: maplet_codomain-mpc_obj_to_be_del +->
nat_ptr\/{null_nat_ptr}
```